Computers and Mathematics with Applications 62 (2011) 2225–2234

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Algorithm for calculating the analytic solution for economic dispatch with multiple fuel units

L. Bayón *, J.M. Grau, M.M. Ruiz, P.M. Suárez

Department of Mathematics, EUITI, University of Oviedo, Gijón 33203, Spain

ARTICLE INFO

Article history:
Received 9 November 2010
Received in revised form 17 March 2011
Accepted 7 July 2011

Keywords:
Economic dispatch
Multiple fuel units
Infimal convolution
Basic recurrence
Divide-and-conquer

ABSTRACT

The problem of economic dispatch with multiple fuel units has been widely addressed via different techniques using approximate methods due to the exponential complexity of full enumeration in the underlying combinatory problem. A method has recently been outlined by Min et al. (2008)[12], that allows the problem to be solved in an exact way in polynomial time. In this paper, we present an alternative technique and take this idea further, studying and comparing two algorithms of polynomial complexity: basic recurrence and divide-and-conquer. Moreover, we provide the exact solution to the problem by Lin and Viviani (1984)[1], that constitutes the traditional test for all approximate methods and present a comprehensive survey of several heuristic approaches.

1. Introduction

This paper presents a method to solve the power economic dispatch (ED) problem with piecewise quadratic cost functions. The ED problem is one of the most important optimization problems in a power system. Economic dispatch is defined as finding an optimal distribution of a system load to the generators in order to minimize the total generation cost while satisfying the total demand and generating capacity constraints. For the sake of simplicity, transmission losses are often omitted with the assumption that P_D accounts for the system loss. Traditionally, the cost function of each generator is approximated by a single quadratic function. The classic ED problem can be described as an optimization (minimization) problem:

$$\text{minimize}: \sum_{i=1}^{N} F_i(P_i) = \sum_{i=1}^{N} \left(\alpha_i + \beta_i P_i + \gamma_i P_i^2\right)$$

$$\text{subject to}: \sum_{i=1}^{N} P_i = P_D; \qquad P_{i\,\min} \le P_i \le P_{i\,\max}, \quad \forall i = 1, \ldots, N$$

where $F_i(P_i)$ is the fuel cost function of the ith unit, P_i is the power generated by the ith unit, P_D is the system load demand, $P_{i\,\min}$ and $P_{i\,\max}$ are the minimum and maximum power outputs of the ith unit and N is the number of dispatchable units. In recent years, however, a considerable number of studies have been conducted on ED with a non-smooth fuel cost function. A common practice in present-day thermal power stations is to use natural gas from multiple gas fields so as to improve the reliability of service in the case of a shortage from any of the gas sources. Other generation units, especially those supplied with numerous sources (gas and oil) of fuel, are faced with the problem of determining which fuel is most economical to burn. For any given unit with multiple cost curves, said curves can be superimposed as in Fig. 1. The resulting cost function is known as the *hybrid* cost function or *piecewise* cost function.

* Corresponding author.
E-mail address: bayon@uniovi.es (L. Bayón).

0898-1221/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2011.07.008

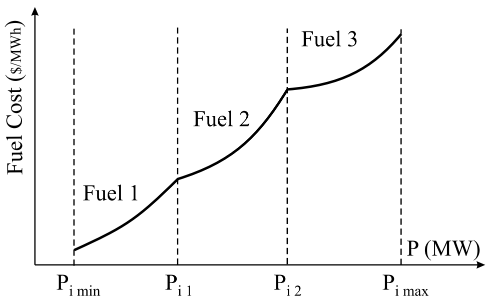

Fig. 1. Hybrid cost function.

These intersecting curves mean that it may be more economical to burn a certain fuel for some MW outputs and another kind of fuel for other outputs. Each segment of the function implies the type of fuel being burned. The *hybrid* cost function can be defined as follows:

$$F_i(P_i) = \begin{cases} \alpha_{i1} + \beta_{i1}P_i + \gamma_{i1}P_i^2 \text{ (fuel 1)} & \text{if } P_{i\,\min} \le P_i \le P_{i1} \\ \alpha_{i2} + \beta_{i2}P_i + \gamma_{i2}P_i^2 \text{ (fuel 2)} & \text{if } P_{i1} \le P_i \le P_{i2} \\ \cdots & \cdots \\ \alpha_{ik} + \beta_{ik}P_i + \gamma_{ik}P_i^2 \text{ (fuel } k\text{)} & \text{if } P_{ik-1} \le P_i \le P_{i\,\max} \end{cases}$$

where α_{ik}, β_{ik} and γ_{ik} are the cost coefficients of the ith generator for type k fuel.

Practical ED problems with multifuel effects are represented as a non-smooth optimization problem with equality and inequality constraints, which makes the problem of finding the global optimum difficult. Many salient methods have been proposed to solve this problem such as a hierarchical method (HM) [1], evolutionary programming (EP) [2], improved evolutionary programming (IEP) [3], Tabu search [4], the Hopfield neural network approach (HNN) [5], the adaptive Hopfield neural network method (AHNN) [6], modified particle swarm optimization (MPSO) [7], the Self-Adaptive Differential Evolution (SDE) algorithm [8], the hybrid real coded genetic algorithm (HGA) [9], a genetic algorithm with multiplier updating (GA_MU) [10] and genetic algorithms for combinatorial optimization problems (GA-COP) [11]. Although these heuristic methods do not always guarantee obtaining the globally optimal solution in finite time, they often provide a fast and reasonable solution (suboptimal nearly global optimal).

Currently, only one systematic approach has been developed to find a global solution to ED with multiple fuel units [12]. Said paper uses the $\lambda - P$ method, a technique based on duality theory. In this paper we present a new technique for solving the ED problem that is based on the calculation of the Infimal Convolution. Furthermore, we study and compare two algorithms of polynomial complexity: basic recurrence and divide-and-conquer, which lead to the determination of the analytic optimal solution. This study constitutes the generalization of prior papers [13–15] in which additional simplifications were considered, such as only including constraints of the type $x_i \ge 0$, or imposing certain conditions in the boundaries of the form: $F_i'(P_{i\,\min}) < F_j'(P_{j\,\max})$, $\forall i, j$.

This paper presents an algorithm for calculating the infimal convolution of N quadratic piecewise functions and is organized as follows. Section 2 provides the necessary mathematical definitions. In Section 3, we first calculate the infimal convolution of two quadratic functions to then extend the result to two piecewise quadratic functions. To finalize, we generalize the result to the case of n piecewise quadratic functions. Section 4 presents a detailed description of the proposed algorithm for calculating the analytic solution. We analyze the computational complexity of the proposed algorithms in Section 5. In Section 6, the proposed method is applied to a classic test ED problem: the 10-generator system of Lin and Viviani [1]. Finally, Section 7 provides a summary of the principal contributions of the paper.

2. Definitions

Definition 1. Let $F, G : \mathbb{R} \longrightarrow \bar{\mathbb{R}}$ be two functions of $\mathbb{R}$ in $\bar{\mathbb{R}} := \mathbb{R} \cup \{+\infty, -\infty\}$. We denote as the Infimal Convolution of F and G the operation defined below:

$$\left(F \bigodot G\right)(x) := \inf_{y \in \mathbb{R}} \{F(x) + G(y - x)\}.$$

It is well known that $\left(\mathcal{F}(\mathbb{R}, \bar{\mathbb{R}}), \bigodot\right)$ is a commutative semigroup. Furthermore, for every finite set $E \subset \mathbb{N}$, it is verified that

$$\left(\bigodot_{i \in E} F_i\right)(K) = \inf_{\sum_{i \in E} x_i = K} \sum_{i \in E} F_i(x_i).$$

When the functions are considered constrained to a particular domain $\mathrm{Dom}(F_i) = [m_i, M_i]$, the above definition continues to be perfectly valid redefining $F_i(x) = +\infty$ if $x \notin \mathrm{Dom}(F_i)$. In this case, the definition may be expressed as follows: Let us denote

$$\left(F_1 \bigodot F_2\right)(\xi) := \min_{\substack{x_1+x_2=\xi \\ m_i \le x_i \le M_i}} (F_1(x_1) + F_2(x_2)) = \min_{\substack{m_1 \le x \le M_1 \\ m_2 \le \xi - x \le M_2}} (F_1(x) + F_2(\xi - x)).$$

This is the abstract functional operation that constructs the cost function of the equivalent thermal power plant to a set of plants with cost functions F_i.

3. Infimal convolution

In this section we shall first analyze the particular case of the infimal convolution of two quadratic functions. This result will form the basis for the subsequent generalization to the case of n functions.

Proposition 1. *Let $F_i(x_i) = \alpha_i + \beta_i x_i + \gamma_i x_i^2$ $(i = 1, 2)$ with domains $[m_i, M_i]$. Let us assume that $F_1'(m_1) \le F_2'(m_2)$.*

(A) *If $F_1'(m_1) \le F_2'(m_2) \le F_1'(M_1) \le F_2'(M_2)$, then:*

$$\left(F_1 \bigodot F_2\right)(\xi) := \begin{cases} F_1(\xi - m_2) + F_2(m_2) & \text{if } \xi \in [m_1 + m_2, m_2 + l_1] \\ F_{12}(\xi) & \text{if } \xi \in [m_2 + l_1, M_1 + l_2] \\ F_2(\xi - M_1) + F_1(M_1) & \text{if } \xi \in [M_1 + l_2, M_1 + M_2] \end{cases}$$

with $l_1 = \frac{(-\beta_1+\beta_2+2\gamma_2 m_2)}{2\gamma_1}$; $l_2 = \frac{(\beta_1-\beta_2+2\gamma_1 M_1)}{2\gamma_2}$.

(B) *If $F_1'(m_1) \le F_2'(m_2) \le F_2'(M_2) \le F_1'(M_1)$, then:*

$$\left(F_1 \bigodot F_2\right)(\xi) := \begin{cases} F_1(\xi - m_2) + F_2(m_2) & \text{if } \xi \in [m_1 + m_2, m_2 + l_1] \\ F_{12}(\xi) & \text{if } \xi \in [m_2 + l_1, M_2 + l_3] \\ F_1(\xi - M_2) + F_2(M_2) & \text{if } \xi \in [M_2 + l_3, M_1 + M_2] \end{cases}$$

with $l_3 = \frac{(-\beta_1+\beta_2+2\gamma_2 M_2)}{2\gamma_1}$.

(C) *If $F_1'(m_1) \le F_1'(M_1) \le F_2'(m_2) \le F_2'(M_2)$, then:*

$$\left(F_1 \bigodot F_2\right)(\xi) := \begin{cases} F_1(\xi - m_2) + F_2(m_2) & \text{if } \xi \in [m_1 + m_2, M_1 + m_2] \\ F_1(M_1) + F_2(\xi - M_1) & \text{if } \xi \in [M_1 + m_2, M_1 + M_2] \end{cases}$$

with

$$F_{12}(\xi) = \alpha_1 + \alpha_2 - \frac{(\beta_1 - \beta_2)^2}{4(\gamma_1 + \gamma_2)} + \frac{\gamma_2\beta_1 + \gamma_1\beta_2}{\gamma_1 + \gamma_2}\xi + \frac{\gamma_1\gamma_2}{\gamma_1 + \gamma_2}\xi^2.$$

Proof. It is simply a question of considering the three possibilities of ordering the values $F_1'(M_1), F_2'(M_2), F_2'(m_2)$, bearing in mind that $F_i'(m_i) < F_i'(M_i)$, $i = 1, 2$ and applying the algorithm elucidated in [15]. □

We shall now see what happens when the cost functions are piecewise quadratic due to presenting a different expression for different ranges. In this case, each of these functions may be understood as the minimum of a family of functions, in the following sense. Let

$$F(x) = \begin{cases} F_1(x) & \text{if } x \in [m_1, M_1] \\ \cdots & \cdots \\ F_k(x) & \text{if } x \in [m_k, M_k] \end{cases}$$

be a piecewise quadratic function. Redefining each

$$F_i(x) := \begin{cases} F_i(x) & \text{if } x \in [m_i, M_i] \\ \infty & \text{if } x \notin [m_i, M_i] \end{cases}$$

for $i = 1, \ldots, k$, we have that $F(x) = \min_{i \in \{1,\ldots,k\}} F_i(x)$. Once redefined in this way, the calculation of the infimal convolution of two piecewise quadratic functions requires a combinatory exploration that is reflected in the following theorem.

Theorem 1. *Let $F(x) := \min_{i \in A}(F_i(x))$ and $G(x) := \min_{i \in B}(G_i(x))$, then:*

$$F \bigodot G = \min_{(i,j) \in A \times B} \left(F_i \bigodot G_j\right).$$

Proof.

$$\begin{aligned}\left(F \bigodot G\right)(t) &= \min_x(F(t-x)+G(x)) = \min_x(\min_{i\in A}(F_i(t-x)) + \min_{j\in B}(G_j(x)))\\ &= \min_x(\min_{(i,j)\in A\times B}(F_i(t-x)+G_j(x)))\\ &= \min_{(i,j)\in A\times B}(\min_x(F_i(t-x)+G_j(x))) = \min_{(i,j)\in A\times B}\left(F_i \bigodot G_j\right)(t). \quad \square\end{aligned}$$

This theorem justifies the construction of the equivalent thermal power plant to two multifuel plants (i.e. the infimal convolution of two piecewise quadratic functions) as the minimum function of all the possible infimal convolutions of pairs of fuels.

Now, bearing in mind the associative nature of the infimal convolution operation, the equivalent of n multifuel plants may be calculated by means of a recursive process, carrying out n operations of infimal convolution.[12] consider the basic recurrence:

$$H_1 \bigodot H_2 \bigodot \cdots \bigodot H_n = \left(H_1 \bigodot H_2 \bigodot \cdots \bigodot H_{n-1}\right) \bigodot H_n.$$

In this paper we shall also consider a divide-and-conquer recurrence:

$$H_1 \bigodot H_2 \bigodot \cdots \bigodot H_n = \left(H_1 \bigodot H_2 \bigodot \cdots \bigodot H_{\frac{n}{2}}\right) \bigodot \left(H_{\frac{n}{2}+1} \bigodot \cdots \bigodot H_n\right).$$

4. Algorithm

In this section we analyze the computational complexity of the two proposed recursive algorithms for calculating the analytic solution. We first analyze the part that is common to both, which it the calculation of the infimal convolution and the calculation of the minimum of a set of polynomial functions. Finally, we discuss the specific aspects of each of the two recursive strategies.

Let F and G be two piecewise quadratic functions:

$$F(x) = \begin{cases} F_1(x) & \text{if } x \in [m_1, M_1] \\ \cdots & \cdots \\ F_k(x) & \text{if } x \in [m_k, M_k]; \end{cases} \qquad G(x) = \begin{cases} G_1(x) & \text{if } x \in [\tilde{m}_1, \tilde{M}_1] \\ \cdots & \cdots \\ G_s(x) & \text{if } x \in [\tilde{m}_s, \tilde{M}_s] \end{cases}$$

considering $F_j(x) := \infty$ if $x \notin [m_j, M_j]$ and $G_j(x) := \infty$ if $x \notin [\tilde{m}_j, \tilde{M}_j]$. Hence,

$$F(x) = \min_{i\in A=\{1,\ldots,k\}} F_i(x) \quad \text{and} \quad G(x) = \min_{i\in B=\{1,\ldots,s\}} G_i(x)$$

The calculation of the infimal convolution $F \bigodot G = \min_{(i,j)\in A\times B}\left(F_i \bigodot G_j\right)$ is carried out in two phases:

PHASE (1) Calculation of $F_i \bigodot G_j$ for each $(i,j) \in A \times B$, which requires, at least, $K_1 \cdot \#A \cdot \#B$ elemental operations, where K_1 represents the number of minimum elemental operations required in the construction of the infimal convolution of two 2nd-order polynomials. Thus, the required running time is $\Omega(\#A \cdot \#B)$ and, likewise, $O(\#A \cdot \#B)$.

PHASE (2) Calculate

$$\min_{(i,j)\in A\times B} F_i \bigodot G_j = \min_{k=1,\ldots,\#A\cdot\#B} P_k.$$

Let $N := \#A \cdot \#B$ and $\Xi_N := \{1, 2, \ldots, N\}$. The functions involved in the proposed algorithms are defined as follows.

(i) Let us consider the function Θ that assigns to each pair of $(i,j) \in \Xi_N^2$ the set of cut-off points of the polynomials P_i and P_j within $[m_i, M_i] \cap [m_j, M_j]$. Note that $\Theta[i,j]$ may have 0, 1 or 2 elements.

$$\Theta[t,i,j] := \{x \in [t,\infty) \cap [m_i, M_i] \cap [m_j, M_j] \text{ such that } P_i(x) = P_j(x)\}.$$

(ii) Let us consider the function B that assigns to each t the subscript of the polynomial whose value at all points of some interval $[t, t+\varepsilon)$ is lower than or equal to the value of the remaining polynomials, defined in said interval, with $t \in [m_{B[t]}, M_{B[t]})$. $B[t]$ satisfies for all j such that $t \in [m_j, M_j)$:

$$t \in [m_{B[t]}, M_{B[t]}), \quad \begin{array}{l} P_{B[t]}(t) \le P_j(t) \\ P_{B[t]}(t) = P_j(t) \Longrightarrow P'_{B[t]}(t) < P'_j(t) \\ P_{B[t]}(t) = P_j(t) \ \text{ and } \ P'_{B[t]}(t) = P'_j(t) \Longrightarrow P''_{B[t]}(t) \le P''_j(t). \end{array}$$

$B[t]$ represents the subscript in whose associated polynomial the minimum searched for in some surrounding $[t, t+\varepsilon)$ is obtained. The number of operations required to determine $B[t]$ is $\Omega(N)$ and $O(N)$ seeing as it actually comprises the search for the minimum element of an ordered set of N elements.

(iii) Let us consider the function C that assigns to each pair $(t, j) \in \mathbb{R} \times (\Xi_N - \{B[t]\})$ the lowest of the points of $[t, \infty) \cap [m_{B[t]}, M_{B[t]}] \cap [m_j, M_j]$ at which the graph of the polynomial $P_{B[t]}$ changes from being below to being above P_j. If this fact is not produced, then we consider $C[t, j] = M_{B[t]}$.

$$\text{If } \Theta[t, B[t], j] = \varnothing \Rightarrow C[t, j] := \begin{cases} m_j & \text{if } m_j \in [t, M_{B[t]}] \wedge P_j(m_j) < P_{B[t]}(m_j) \\ M_{B[t]} & \text{otherwise} \end{cases}$$

$$\text{If } \Theta[t, B[t], j] \neq \varnothing \Rightarrow C[t, j] := \begin{cases} M_{B[t]} & \text{if } P_{B[t]} = P_j \\ m_j & \text{if } m_j \in [t, M_{B[t]}] \wedge P_j(m_j) < P_{B[t]}(m_j) \\ \min(\Theta[t, B[t], j]) & \text{otherwise.} \end{cases}$$

(iv) Let us now consider the function $H : \mathbb{R} \to \Xi_N$

$$H[t] := \{C[t, j] \mid j \in \{1, \ldots, N\} - \{B[t]\}\}$$

that returns the set of points resulting from the action of the function $C[t, \cdot]$. The number of operations needed to determine $H[t]$ is also $\Omega(N)$ and $O(N)$.

4.1. Description of the algorithm

Let us represent each polynomial $P_i(x) = \alpha_i + \beta_i x + \gamma_i x^2$ restricting the domain $[m_1, M_1]$ by means of the list: $\{m_1, M_1, \alpha_1, \beta_1, \gamma_1\}$.

$$\textit{Input}: \begin{cases} \{\{m_1, M_1, \alpha_1, \beta_1, \gamma_1\}, \ldots, \{m_N, M_N, \alpha_N, \beta_N, \gamma_N\}\} \\ \text{Aux} = \{\}; \quad t_1 = \min\limits_{i=1,\ldots,N} \{m_i\} \end{cases}$$

IF $t_s = \max\{M_i\}$ then STOP

ELSE $\begin{array}{l} t_{s+1} := \min H[t_s] \\ \text{Aux} = \text{Join}[\text{Aux}, \{\{t_s, t_{s+1}, \alpha_{B[t_s]}, \beta_{B[t_s]}, \gamma_{B[t_s]}\}\}] \end{array}$

Output : $\text{Aux} = \{\{t_1, t_2, \alpha_{B[t_1]}, \beta_{B[t_1]}, \gamma_{B[t_1]}\}, \ldots, \{t_s, t_{s+1}, \alpha_{B[t_s]}, \beta_{B[t_s]}, \gamma_{B_{[t_s]}}\} \ldots\}$

The solution is Aux, which represents the piecewise quadratic function:

$$\begin{cases} \alpha_{B[t_1]} + \beta_{B[t_1]} x + \gamma_{B[t_1]} x^2 & \text{if } x \in [t_1, t_2] \\ \cdots & \cdots \\ \alpha_{B[t_s]} + \beta_{B[t_s]} x + \gamma_{B[t_s]} x^2 & \text{if } x \in [t_s, t_{s+1}]. \end{cases}$$

Thus, the running time of the algorithm depends linearly on the number of recursive calls, which is, in short, the length of the output list: the number of different intervals involved in the definition of the piecewise quadratic function solution.

Let us denote by $\Phi(k)$ the number of fuels present in the infimal convolution $H_1 \bigodot H_2 \bigodot \cdots \bigodot H_k$ and, for the sake of convenience, let us assume that all power plants have an identical number of fuels: let us say η. We shall now analyze the number of operations needed in the two recursive strategies under these conditions.

5. Computational complexity

We propose two recursive strategies:

5.1. Basic recurrence

Let us see the number of operations needed to perform the recursive loop

$$\left(H_1 \bigodot H_2 \bigodot \cdots \bigodot H_{n-1}\right) \bigodot H_n.$$

PHASE (1) It is easily shown that the number of elemental operations s_1 satisfies:

$$K_1 \Phi(n-1) * \eta \leq s_1 \leq K_2 \Phi(n-1) * \eta$$

and, in short, that $s_1 \in \Omega(\Phi) \cap O(\Phi)$.

PHASE (2) It is easily shown that the number of elemental operations s_2 satisfies:

$$K_1 \Phi(n-1) * \eta * \Phi(n) \leq s_2 \leq \Phi(n-1) * \eta * \Phi(n)$$

and, in short, that $s_2 \in \Omega(\Phi^2) \cap O(\Phi^2)$.

Hence, the total number of operations of the recursive loop is $s = s_1 + s_2$, and the resulting recursive equation is:

$$T(n) = T(n-1) + s(n) \quad \text{with } s \in \Omega(\Phi^2)$$

5.2. Divide-and-conquer

Let us see the number of operations needed to perform the recursive loop

$$\left(H_1 \bigodot H_2 \bigodot \cdots \bigodot H_{\frac{n}{2}}\right) \bigodot \left(H_{\frac{n}{2}+1} \bigodot \cdots \bigodot H_n\right).$$

PHASE (1) It is easily shown that the number of elemental operations $\hat{s}_1$ satisfies:

$$K_1 \left(\Phi\left(\frac{n}{2}\right)\right)^2 \le \hat{s}_1 \le K_2 \left(\Phi\left(\frac{n}{2}\right)\right)^2.$$

Such that $\tilde{s}_1 \in \Omega(\Phi^2) \cap O(\Phi^2)$.

PHASE (2) It is easily shown that the number of elemental operations $\hat{s}_2$ satisfies:

$$K_1 \Phi(n) * \left(\Phi\left(\frac{n}{2}\right)\right)^2 \le \hat{s}_2 \le K_2 \Phi(n) * \left(\Phi\left(\frac{n}{2}\right)\right)^2$$

and, in short, that $\tilde{s}_2 \in \Omega(\Phi^3) \cap O(\Phi^3)$.

Hence, the total number of operations of the recursive loop is $\hat{s} = \hat{s}_1 + \hat{s}_2$, and the resulting recursive equation is:

$$\tilde{T}(n) = 2\tilde{T}\left(\frac{n}{2}\right) + \tilde{s}(n) \quad \text{with } \tilde{s} \in \Omega(\Phi^3).$$

5.3. Results

Theorem 2. *The computational complexity of the basic recurrence and of divide-and-conquer is at least cubic in order; i.e.* $T, \tilde{T} \in \Omega(n^3)$.

Proof. It is easily shown that the function Φ is, at least, linear in order ($\Phi \in \Omega(n)$); bear in mind that the case that provides a lower value is produced when the cost functions are defined in a single range (one single fuel), this value being exactly $2n - 1$ (see [15]). Hence:

$$T(n) = T(n-1) + s(n) \quad \text{with } s \in \Omega(n^2)$$
$$\tilde{T}(n) = 2\tilde{T}\left(\frac{n}{2}\right) + \tilde{s}(n) \quad \text{with } \tilde{s} \in \Omega(n^3).$$

Now, it is also well known that from these recursive relations, it follows that $T \in \Omega(n^3)$ and $\tilde{T} \in \Omega(n^3)$. □

Remark 1. It should be pointed out here that, although [12] assures (without providing any justification) that the running time of the basic recurrence is linear, in view of their analysis of the Lin and Viviani test [1], this may not be so even in the most favorable case. A recursive algorithm (basic recurrence) can only be linear in order ($O(n)$) if the order of complexity of the loop operation with the recursive call is constant ($T(n) = T(n-1) + O(1)$). This is obviously impossible as the output of the algorithm in step $(n-1)$ is a piecewise-defined function that involves a number of pieces $\Phi(n)$ and which is at least linear in order. Therefore, simply reading the data related to each piece would be of this same order. Hence

$$T(n) = T(n-1) + O(n) => T \in O(n^2).$$

Quadratic order is therefore the maximum aspiration even when conceiving of the most efficient algorithm possible. The one we propose here, besides reading the output data from step $(n-1)$, performs a total number of operations that is quadratic in order and which situates the algorithm in cubic time. Although our algorithm could be further developed taking advantage of certain peculiarities of the nature of the problem, we have serious doubts as to whether this development could achieve a quadratic order of complexity and, as already stated, a linear order of complexity in any case whatsoever. Naturally, Lin and Viviani's test involves a small number of power plants and may give the impression that the times grow linearly. However, the growth is of course asymptotically at least polynomial in order.

Remark 2. It is obvious that the running time of both algorithms ultimately depends on the function $\Phi(n)$. Although it may appear that $\Phi(n)$ might be exponential in order in the worst case (all combinations of fuels), it can immediately be seen that highly disparate combinations of fuels are not present in the infimal convolution, as they disappear from the algorithm in Phase (B) (calculation of the minimum). Multiple experiments provide support for the idea that the growth of Φ is linear in order, which allows us to make the following conjecture.

Conjecture 1. *The running times of* T *and* $\tilde{T}$ *satisfy* $\tilde{T} \in O(n^3)$ *and* $T \in O(n^3)$.

Remark 3. Even when considering that both strategies (basic recurrence and divide-and-conquer) have an identical order of complexity, $O(n^3)$, we might be led to think that a high number of η (number of power plant fuels) might mean that $T(n)$ are dominated by $\tilde{T}(n)$. However, a high value of η leads to the value of $\Phi(n)$ likewise being increased, which in the divide-and-conquer algorithm appears raised to the cube. Thus, it is asymptotically reasonable for the basic recurrence to be more efficient than the divide-and-conquer strategy, and it is in fact so. Nevertheless, for relatively high values of η, divide-and-conquer is preferable to basic recurrence provided that the number of power plants involved is moderately low.

Table 1
Exact solution.

Pmin–Pmax	a	b	c	Fuels
1353.00–1361.32	0.002176	−5.8506	4138.03	1211121131
1361.32–1406.19	0.000747	−1.9605	1490.21	1211121131
1406.19–1415.43	0.000336	−0.8053	678.01	1211121131
1415.43–1473.07	0.000255	−0.574	514.3	1211121131
1473.07–1540.50	0.000171	−0.3286	333.57	1211121131
1540.50–1553.41	0.000341	−0.8819	784.36	1211121111
1553.41–1616.44	0.000207	−0.4651	460.61	1211121111
1616.44–1672.09	0.000218	−0.5107	503.49	1311121111
1672.09–1697.25	0.000251	−0.6366	623.57	1312121111
1697.25–1715.87	0.000238	−0.5997	597.37	1111121111
1715.87–1735.09	0.000294	−0.8062	786.24	1312121211
1735.09–1754.01	0.000277	−0.7533	746.47	1111121211
1754.01–1772.92	0.000356	−1.0472	1018.38	1312111211
1772.92–1798.63	0.000331	−0.9671	954.85	1111111211
1798.63–1889.87	0.000412	−1.2847	1265.75	1112111211
1889.87–1950.78	0.000321	−0.942	941.86	1112111211
1950.78–1962.54	0.000249	−0.6605	667.31	1112111211
1962.54–2002.55	0.000342	−1.0606	1092.55	1113111211
2002.55–2015.36	0.000261	−0.7368	768.36	1112131211
2015.36–2054.32	0.000367	−1.1961	1266.73	1112131311
2054.32–2067.85	0.000275	−0.8213	881.74	1112131311
2067.85–2068.21	0.000219	−0.5874	639.93	1112131311
2068.21–2106.10	0.000395	−1.3524	1469.79	1113131311
2106.10–2118.90	0.000291	−0.9154	1009.55	1113131311
2118.90–2170.67	0.000229	−0.6509	729.41	1113131311
2170.67–2436.35	0.000189	−0.4805	544.38	1113131311
2436.35–2614.10	0.000187	−0.4701	535.38	2113131311
2614.10–2744.62	0.000157	−0.3439	404.46	2113131331
2744.62–2881.66	0.000212	−0.6434	815.36	2113131331
2881.66–2918.90	0.000214	−0.6656	862.46	2113132331
2918.90–2959.07	0.000177	−0.5244	769.94	2123131331
2959.07–3077.97	0.000248	−0.948	1396.64	2123131331
3077.97–3161.48	0.000251	−0.9775	1460.88	2123132331
3161.48–3189.09	0.000272	−1.1269	1720.39	2123232331
3189.09–3290.50	0.000139	−0.3125	479.64	2123133331
3290.50–3303.30	0.000320	−1.5034	2438.99	2123133331
3303.30–3318.75	0.000145	−0.3621	574.83	2123233331
3318.75–3351.54	0.000355	−1.7567	2889.05	2123233331
3351.54–3385.51	0.000439	−2.3185	3830.56	2123233331
3385.51–3462.58	0.000159	−0.4989	881.77	2123133332
3462.58–3474.40	0.000451	−2.5191	4379.23	2123133332
3474.40–3490.84	0.000167	−0.5657	1013.94	2123233332
3490.84–3513.06	0.000524	−3.059	5365.65	2123233332
3513.06–3537.47	0.000730	−4.5036	7903.28	2123233332
3537.47–3567.33	0.000884	−5.5917	9827.69	2123233332
3567.33–3607.37	0.000083	−0.0066	91.81	2123333332
3607.37–3629.59	0.000187	−0.754	1439.78	2123333332
3629.59–3644.51	0.000780	−5.0644	9262.24	2123333332
3644.51–3657.77	0.001344	−9.1736	16 750.4	2123333332
3657.77–3695.00	0.001978	−13.8123	25 234.0	2123333332

6. Example 1

The proposed method is applied to a classic test ED problem: Lin and Viviani's 10-generator system [1]. The fuel cost data of the generators is given in said paper. This example has been tested by numerous authors: [10,8,7]. The optimal algorithm was implemented on a personal computer (Pentium IV, 3.4 GHz PC) in Mathematica 5.0. The analytic solution of our method is compared with the best results of other methods.

6.1. Exact solution

The results of the proposed algorithm are summarized in Table 1. This table shows the fuel cost functions and combinations of fuel types for a power range of 1353 MW–3695 MW, which represent the minimum to maximum feasible generating power in the system, respectively. The global solutions to the 10-generator system can be easily obtained with our method.

Table 2
Comparison with other methods.

U	HM		HNN		AHNN		EP		IEP		MPSO	
	F	GEN	F	GEN	F	GEN	F	GEN	F	GEN	F	GEN
1	2	218.4	2	224.5	2	225.7	2	225.2	2	219.5	2	218.3
2	1	211.8	1	215.0	1	215.2	1	215.6	1	211.4	1	211.7
3	1	281.0	3	291.8	1	291.8	1	291.8	1	279.7	1	280.7
4	3	239.7	3	242.2	3	242.3	3	242.1	3	240.3	3	239.6
5	1	279.0	1	293.3	1	293.7	1	293.7	1	276.5	1	278.5
6	3	239.7	3	242.2	3	242.3	3	241.9	3	239.9	3	239.6
7	1	289.0	1	303.1	1	302.8	1	301.6	1	289.0	1	288.6
8	3	239.7	3	242.2	3	242.3	3	242.8	3	241.3	3	239.6
9	3	429.2	1	335.7	1	355.1	1	356.6	3	425.1	3	428.5
10	1	275.2	1	289.5	1	288.8	1	288.7	1	277.2	1	274.9
TC	625.18		626.12		626.24		626.26		623.851		623.809	
U	HGA		SDE		IGA_MU		CGA_MU		GA-COP		EXACT	
	F	GEN	F	GEN	F	GEN	F	GEN	F	GEN	F	GEN
1	2	218.2559	2	218.2499	2	218.12	2	218.46	2	218.25	2	218.249911
2	1	211.6816	1	211.6626	1	211.68	1	211.51	1	211.66	1	211.662633
3	1	280.7359	1	280.7228	1	280.86	1	280.90	1	280.72	1	280.722776
4	3	239.6298	3	239.6315	3	239.65	3	239.62	3	239.63	3	239.631522
5	1	278.4819	1	278.4973	1	278.63	1	278.50	1	278.50	1	278.497265
6	3	239.6508	3	239.6315	3	239.61	3	239.64	3	239.63	3	239.631522
7	1	288.5721	1	288.5845	1	288.57	1	288.62	1	288.58	1	288.584539
8	3	239.6280	3	239.6315	3	239.71	3	239.62	3	239.63	3	239.631522
9	3	428.5175	3	428.5216	3	428.45	3	428.58	3	428.52	3	428.521622
10	1	274.8466	1	274.8667	1	274.70	1	274.55	1	274.87	1	274.866683
TC	623.8092		623.8091		623.8093		623.8095		623.8092		623.809154	

6.2. Comparison with other techniques

The proposed algorithm is applied to the ED problem with the 10-generator system. During the test, the total system demand is fixed at 2700 MW.

The exact solution is compared with the results of various heuristic approaches including HM [1], HNN [5], AHNN [6], EP [2], IEP [3], MPSO [7], HGA [9], SDE [8], the improved genetic algorithm with multiplier updating (IGA_MU) and conventional (CGA_MU) [10], and GA-COP [11]. Table 2 shows the best optimal dispatch results obtained using these other methods.

As can be seen, the best approximate methods are found to be SDE and GA-COP. Their similarity to the exact solution can be appreciated in Table 2.

7. Example 2

In this section, the proposed algorithm is applied to the real IEEE 30-bus system [16,17] with piecewise quadratic cost functions for generating units. Since our study, and all those inspired by the model of Lin and Viviani [1], do not consider transmission losses, line data and the optimal power load flow are neglected in this paper. The fuel cost data of the generators is given in [17]. Table 3 presents the analytic solution of our method for this test.

We next applied our algorithm to other systems resulting from carrying out several replicas in the IEEE 30-bus system. Table 4 shows the running times of our algorithm and $\Phi(n)$, the number of fuels present in the infimal convolution.

The running times of these examples show that the computational complexity of our algorithm is near to the conjectured cubic order. This test also provides support for the idea that the growth of $\Phi(n)$ is moderate (less than to linear in order), as we conjectured in Remark 2. The value of τ, using the slow symbolic package Mathematica and a personal computer (Pentium IV, 3.4 GHz PC) is 11.2 s. We estimate that using a programming language like C++ will provide us results less than to 0.1 s.

8. Conclusions

In this paper we have presented a method that provides the exact solution to the Economic Dispatch problem with multiple fuel units that is based on the concept of infimal convolution. In addition, two algorithms of polynomial complexity are proposed to calculate the analytic solution and are compared. Using Lin and Viviani's ED test, our solution is compared with the results of various heuristic approaches.

Besides surpassing any heuristic-type algorithm as regards precision for a specific problem, the proposed algorithm simultaneously solves the family of problems resulting from considering all the possible levels of power demand. For this reason, it would be perfectly applicable to real situations in real time.

Table 3
Exact solution.

*P*min–*P*max	*a*	*b*	*c*	Fuels
117.00–117.04	876.797	−12.375	0.06250	111111
117.04–135.64	66.664	1.4687	0.00336	111111
135.64–176.64	56.466	1.6190	0.00280	111111
176.64–187.12	385.969	−2.1116	0.01336	111111
187.12–191.33	2106.384	−20.500	0.06250	111111
191.33–196.99	87.703	0.6011	0.00736	111111
196.99–203.96	386.635	−2.1602	0.01367	121111
203.98–210.36	33.389	1.3038	0.00518	121111
210.36–279.74	78.582	1.4325	0.00355	211111
279.74–280.33	−4.227	2.0246	0.00249	211111
280.33–292.36	52.684	1.6925	0.00295	221111
292.36–295.12	−13.162	2.1429	0.00218	221111
295.12–301.99	248.245	0.3713	0.00518	221111
301.99–307.67	446.875	−0.9441	0.00736	221111
307.67–307.80	5666.609	−34.8750	0.06250	221111
307.80–312.00	1437.107	−7.3929	0.01786	221111
312.00–330.00	712.822	−2.7500	0.01042	221111
330.00–335.00	939.697	−4.1250	0.01250	221111
335.00–340.00	2342.509	−12.5000	0.02500	221111
340.00–347.80	497.467	−1.3567	0.00826	221211
347.80–350.00	2522.279	−13.0000	0.02500	221211
350.00–365.00	3568.529	−19.4500	0.03500	221212
365.00–380.00	2403.529	−12.0500	0.02350	231211
380.00–390.86	1167.664	−5.1808	0.01406	231212
390.86–395.00	4366.679	−21.5500	0.03500	231212
395.00–400.00	4237.929	−21.2000	0.03500	231221
400.00–405.00	1790.742	−8.0750	0.01750	231222
405.00–415.00	4661.179	−22.2500	0.03500	231222
415.00–423.94	3060.848	−14.0362	0.02457	232222
423.94–425.00	13 471.492	−63.1500	0.08250	232222
425.00–433.94	3293.931	−14.5277	0.02457	332222
433.94–435.00	14 201.506	−64.8000	0.08250	332222

Table 4
Running times and $\Phi(n)$.

IEEE 30-bus system	Time (s)	$\Phi(n)$
1×	τ	32
2×	8.8τ	47
3×	33.2τ	60
4×	90.5τ	73
5×	193.1τ	80

Bear in mind that the algorithm would first run with the technical data from the power stations, regardless of the power demand value. The definitive solution of the problem for a specific demand would then become trivial, seeing as it would require no more than the determination of the interval in which said power level is situated. Once this interval has been determined, it would only remain to proceed to ''share out'' the demand among the different power stations, each using the type of fuel that corresponds to it in accordance with the solution provided by the algorithm. This sharing out requires no more than solving an extremely simple problem of separable quadratic programming.

In summary, our algorithm solves in an exact way, for the different ranges of power demand, the underlying combinatorial problem of determining the fuel that must be used by each power station and the power generation cost curve. The problem thus becomes a very simple one of separable quadratic programming that may be addressed approximately using different techniques [18] or, once again, exactly, by means of a quasilinear algorithm as presented in [15] or [19].

At the same time, the present paper opens up an interesting line of theoretical study concerning the infimal convolution operator of piecewise convex functions and, more especially, the computational complexity of its calculation in the case of piecewise quadratic functions.

Acknowledgments

The work of the first author was supported by the Government of Principality of Asturias through PCTI: FICYT IB09-085 and by the Spanish Government (MICINN, project: MTM2010-15737).

References

[1] C.E. Lin, G.L. Viviani, Hierarchical economic dispatch for piecewise quadratic cost functions, IEEE Trans. Power Appar. Syst. 103 (6) (1984) 1170–1175.
[2] T. Jayabarathi, G. Sadasivam, Evolutionary programming-based economic dispatch for units with multiple fuel options, Eur. Trans. Electr. Power 10 (3) (2000) 167–170.
[3] Y.M. Park, J.R. Won, J.B. Park, A new approach to economic load dispatch based on improved evolutionary programming, Eng. Intell. Syst. Electr. Eng. Commun. 6 (2) (1998) 103–110.
[4] W.M. Lin, F.S. Cheng, M.T. Tsay, An improved Tabu search for economic dispatch with multiple minima, IEEE Trans. Power Syst. 17 (2002) 108–112.
[5] J.H. Park, Y.S. Kim, I.K. Eom, K.Y. Lee, Economic load dispatch for piecewise quadratic cost function using hopfield neural network, IEEE Trans. Power Syst. 8 (3) (1993) 1030–1038.
[6] K.Y. Lee, A. Sode-Yome, J.H. Park, Adaptive hopfield neural network for economic load dispatch, IEEE Trans. Power Syst. 13 (2) (1998) 519–526.
[7] J.B. Park, K.S. Lee, J.R. Shin, K.Y. Lee, A particle swarm optimization for economic dispatch with nonsmooth cost functions, IEEE Trans. Power Syst. 20 (1) (2005) 34–42.
[8] R. Balamurugan, S. Subramanian, Self-adaptive differential evolution based power economic dispatch of generators with valve-point effects and multiple fuel options, Int. J. Comput. Sci. Eng. 1 (1) (2007) 10–17.
[9] S. Baskar, P. Subbaraj, M.V.C. Rao, Hybrid real coded genetic algorithm solution to economic dispatch problem, Comput. Electr. Eng. 29 (2003) 407–419.
[10] C. Chiang, Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels, IEEE. Trans. Power Syst. 20 (4) (2005) 1690–1699.
[11] K.I. Min, S.W. Lee, Y.H. Moon, An economic dispatch algorithm as combinatorial optimization problems, Int. J. Control Autom. Syst. 6 (4) (2008) 468–476.
[12] K.I. Min, S.W. Lee, Y.H. Moon, A global solution to economic dispatch with multiple fuel units using a function merger, in: Proceedings of the 17th World Congress IFAC Seoul, Korea, 2008, pp. 11064–11069.
[13] L. Bayón, J.M. Grau, P.M. Suárez, A new formulation of the equivalent thermal in optimization of hydrothermal systems, Math. Probl. Eng. 8 (3) (2002) 181–196.
[14] L. Bayón, J.M. Grau, M.M. Ruiz, P.M. Suárez, New developments on equivalent thermal in hydrothermal optimization: an algorithm of approximation, J. Comput. Appl. Math. 175 (1) (2005) 63–75.
[15] L. Bayón, J.M. Grau, M.M. Ruiz, P.M. Suárez, An analytic solution for some separable convex quadratic programming problems with equality and inequality constraints, J. Math. Inequal. 4 (3) (2010) 453–465.
[16] V.N. Dieu, W. Ongsakul, Enhanced augmented lagrange hopfield network for economic dispatch with piecewise quadratic cost functions, in: Proceedings of Energy for Sustainable Development: Prospects and Issues for Asia, Phuket, March 1–3, 2006.
[17] K.Y. Lee, F.M. Nuroglu, A. Sode Yome, Real Power optimization with load flow using adaptive hopfield neural networks, Eng. Intell. Syst. 8 (1) (2000) 53–58.
[18] N.I.M. Gould, PH.L. Toint, A quadratic programming bibliography, http://www.optimization-online.org/DB-HTML/2001/02/285.html.
[19] L. Bayón, J.M. Grau, M.M. Ruiz, P.M. Suárez, A quasi-linear algorithm for calculating the infimal convolution of convex quadratic functions, in: J. Vigo-Aguiar (Ed.), Proc. CMMSE 2010, vol. I, 2010, pp. 169–172.