Chinese Journal of Chemical Engineering 22 (2014) 742–747

Contents lists available at ScienceDirect

Chinese Journal of Chemical Engineering

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

Process Systems Engineering

Genetic Algorithm Based on Duality Principle for Bilevel Programming Problem in Steel-making Production ☆

Shuo Lin [1,*], Fangjun Luan [1], Zhonghua Han [1], Xisheng Lü [2], Xiaofeng Zhou [2], Wei Liu [3]

[1] Information & Control Engineering Faculty, Shenyang Jianzhu University, Shenyang 110168, China
[2] Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
[3] School of Information Science & Engineering, Northeastern University, Shenyang 110004, China

ARTICLE INFO

Article history:
Received 27 December 2013
Received in revised form 19 February 2014
Accepted 3 March 2014
Available online 25 June 2014

Keywords:
Steel-making
Genetic algorithm
Bilevel problem
Scheduling

ABSTRACT

Steel-making and continuous/ingot casting are the key processes of modern iron and steel enterprises. Bilevel programming problems (BLPPs) are the optimization problems with hierarchical structure. In steel-making production, the plan is not only decided by the steel-making scheduling, but also by the transportation equipment. This paper proposes a genetic algorithm to solve continuous and ingot casting scheduling problems. Based on the characteristics of the problems involved, a genetic algorithm is proposed for solving the bilevel programming problem in steel-making production. Furthermore, based on the simplex method, a new crossover operator is designed to improve the efficiency of the genetic algorithm. Finally, the convergence is analyzed. Using actual data the validity of the proposed algorithm is proved and the application results in the steel plant are analyzed.

1. Introduction

During recent years, the continuous casting technology developed in the 1950s underwent a rapid growth. Because the continuous casting technology can save more energy consumption in the production process from hot steel to slab, the main steels are produced by continuous casting. During the process of steel-making and continuous casting, most of the material transportations within the whole process are finished by cranes. The crane scheduling is the scheduling important branch to assist the scheduling of the main equipments (steel-making, refining, continuous casting machines). A coordinating steel-making and continuous casting system consists of the scheduling of main equipments and assist transportation equipments. Because a good iron and steel making scheduling system could not only realize the reduction of the energy but also improve the production, and the steel-making and continuous casting is the bottleneck of the whole iron and steel production, the crane scheduling plays an important role in the whole process. The major problem of this paper is how to make a good crane schedule in a computationally efficient manner and effectively assist the scheduling of the main equipments in order to ensure a well-organized rhythm in the whole production and to improve the transportation efficiency of the cranes.

Harjunkoski and Grossmann [1] developed a decomposition strategy for solving large scheduling problems using mathematical programming methods. Lee *et al.* [2] solved a scheduling problem for operating the continuous caster by using the concept of a special class of graphs known as interval graphs. Ouelhadj *et al.* [3] described a new model for robust predictive/reactive scheduling of SCC based on the use of multi-agents, tabu search and heuristic approaches. At Stahl Linz GmbH, Neuwirth [4] reported a linear programming model with machine convicts and provided key modeling factors of SCC scheduling and charge allocation scheme in the furnace, but the mathematical representation of the model was not given. An expert system was used by Jimichi *et al.* [5] to determine parameters and operational conditions to match slab production with customer orders.

Stahl Linz GmbH, Stohl and Spopek [6] established a hybrid co-operative expert system model for the SCC scheduling problems, but they were unable to construct an optimized mathematical model. Ferretti *et al.* [7] presented the algorithmic solution based on an ant system metaheuristic for an industrial production inventory problem in a steel continuous casting plant. The model took into account the relevant parameters of the finite-capacity production system. The study focused on the optimization of the production sequence of the billets. Blackburn and Millen [8] examined the impact of a rolling-schedule implementation on the performance of three of the better known lot-sizing methods for single-level assembly systems. Atighehchian *et al.* [9] developed a novel iterative algorithm for scheduling steel-making continuous casting production, named HANO. This algorithm was based on a combination of ant colony optimization (ACO) and non-linear optimization methods. Zhu *et al.* [10] set a novel optimization model to improve the efficiency and performance for production planning in steelmaking and continuous casting (SCC) process.

☆ Supported by the Educational Commission of Liaoning Province Science and Technology Research Projects (L2013237).
* Corresponding author.
E-mail address: farewell_lin@163.com (S. Lin).

http://dx.doi.org/10.1016/j.cjche.2014.05.006
1004-9541/

2. Genetic Algorithm for Bilevel Problem

Bilevel-programming techniques were initiated by Von Stackelberg [11] for bilevel programming problems (BLPPs) based on the dual programming of the follower's problem and the dual theorem. The region of the leader's variables was divided into several sub-regions, such that for all values of the leader's variables in each sub-region, the follower's problem had the same solution expression. Furthermore, a genetic algorithm was used to solve the leader's problems, in which a new crossover operator was designed based on the simplex method. An advantage of the proposed genetic algorithm was that the optimal solutions to the follower's problem can be obtained very easily without massive calculation of the follower's problem. Also, a method of dynamic penalty function was presented.

2.1. Bilevel problem

The bilevel problem is as follows:

$$\begin{cases} \min\limits_{x\in X} F(x,y) \\ \text{s.t.}G(x,y)<0 \\ \min\limits_{y} dy \qquad \text{where } \boldsymbol{d}^{\mathrm{T}}\in\boldsymbol{R}^{n}, \boldsymbol{A}\in\boldsymbol{R}^{q\times m}, \boldsymbol{B}\in\boldsymbol{R}^{q\times n}, \boldsymbol{C}\in\boldsymbol{R}^{q}. \\ \text{s.t.}\boldsymbol{A}x+\boldsymbol{B}y\le\boldsymbol{C}, y\ge 0 \end{cases} \tag{1}$$

The dual problem is as follows:

$$\begin{cases} \min\limits_{u} (\boldsymbol{C}-\boldsymbol{A}x)^{\mathrm{T}}u \\ \text{s.t.}\boldsymbol{B}^{\mathrm{T}}u>-\boldsymbol{d}^{\mathrm{T}}, u>0. \end{cases} \tag{2}$$

$\{u_1, u_2, \ldots, u_k\}$ is the basic feasible solution of problem (2). $\{\boldsymbol{B}_1, \boldsymbol{B}_2, \ldots, \boldsymbol{B}_k\}$ is the feasible basis. $(\boldsymbol{C}-\boldsymbol{A}x)^{\mathrm{T}}_{\boldsymbol{B}_i}$ is the corresponding objective function of $\boldsymbol{B}_i$. The necessary and sufficient condition that $\boldsymbol{B}_i$ is the optimal feasible basis is $(\boldsymbol{C}-\boldsymbol{A}x)^{\mathrm{T}}-(\boldsymbol{C}-\boldsymbol{A}x)^{\mathrm{T}}_{\boldsymbol{B}_i}\boldsymbol{B}_i^{-1}\boldsymbol{B}^{\mathrm{T}}\ge 0, i=1,2,\ldots,k$. For each i,inequality i is $(\boldsymbol{C}-\boldsymbol{A}x)^{\mathrm{T}}-(\boldsymbol{C}-\boldsymbol{A}x)^{\mathrm{T}}_{\boldsymbol{B}_i}\boldsymbol{B}_i^{-1}\boldsymbol{B}^{\mathrm{T}}\ge 0$. So there are k inequality groups. Domain i is the divided domain of inequality i. If leader variable x satisfies inequality i, it belongs to domain i. According to the duality principle, the optimal solution of the follower problem is $(\boldsymbol{C}-\boldsymbol{A}x)^{\mathrm{T}}_{\boldsymbol{B}_i}\boldsymbol{B}_i^{-1}$. There are two steps in the calculation.

(1) Solve the domain i of leader variable x, calculate $y(x)=\left((\boldsymbol{C}-\boldsymbol{A}x)^{\mathrm{T}}_{\boldsymbol{B}_i}\boldsymbol{B}_i^{-1}\right)^{\mathrm{T}}$;
(2) Calculate the problem:

$$\begin{cases} \min\limits_{x\in \boldsymbol{X}} F(x,y(x)) \\ \text{s.t.}G(x,y(x))<0. \end{cases} \tag{3}$$

2.2. Fitness function

Set $R(x)=F(x,y(x))+M_g$ and $\max\{G(x,y(x)),0,i=1,2,\ldots,p\}$ is individual fitness. Where g is the iteration of genetic; $G(x, y(x))$ is the component of $G(x,y(x))$, M_g is the dynamic penalty factor and M_1 is a pre-given positive integer decided by the point in population. Set $x_0, x_1, x_2, \ldots, x_n$ is $n+1$ optimal and no identical points which are collected by fitness. Calculate

$$d=\frac{1}{n+1}\sum_{i=1}^{n}\|x_i-x_G\| \quad \text{where} \quad x_G=\frac{1}{n+1}\sum_{i=0}^{1}x_i \tag{4}$$

$$M_{g+1}=\max\left\{M_g,\frac{1}{d}\right\}. \tag{5}$$

If $n+1$ points are affine independence, $\{x_0, x_1, x_2, \ldots, x_n\}$ is a simplex. With an increasing iteration number, points in population tend to accordance, when M_g tends to infinity.

2.3. Crossover operator

p_s is population size. Sort the fitness of pop(k) points ascending as $x_1, x_2, \ldots, x_{p_s}$. If $x_1, x_2, \ldots, x_{i_0}(i_0<n)$ is affine independence, set $S=\{x_1, x_2, \ldots, x_{i_0}\}$. If point x_{i_0+1} and points in S is affine independence, $S=S\cup\{x_{i_0+1}\}$; else study to the next point x_{i_0+2} until that S includes n points. If there are not n affine independence points, x_1 is the vertex and construct other $n-1$ points in its neighborhood. Make sure n points are affine independence; The construction method is from Takahama T. and Sakai S. [11]. Re-sort the population, delete the worst $n-1$ points, make p_s unchanged. Calculate the center of gravity: $\hat{x}_G=\frac{1}{n}\sum\limits_{s_i\in S}s_i$, x_s is the last point in S, and $\hat{S}=\{x|R(x)\le R(x_s), x\in\text{pop}(k)\}$, so $S\subseteq\hat{S}$, Offspring of crossover individual x,is

$$o_c=\begin{cases} x+r\delta(\hat{x}_G-x), x\notin\hat{S} \\ x+r\Delta(x_1-x), x\in\hat{S} \end{cases} \tag{6}$$

where, $r\in[0,1]$ is random number, δ and Δ is positive constant. When $r=1$, o_c reaches boundary of search field. x_1 is the best point in population. When crossover individual x and points in S is affine independence, there is a simplex which can use the simplex algorithm.

2.4. Mutation operator

$\hat{x}$ is the father individual:

If $\|x_1-\hat{x}\|\le\varepsilon$, mutation offspring $o_m=\hat{x}+\Delta_1$; else $o_m=\hat{x}+\text{diag}(\Delta_2)(x_1-\hat{x})$;

where $\Delta_i=(\Delta_{i1},\Delta_{i2},\ldots,\Delta_{in})^{\mathrm{T}}, i=1,2$; $\Delta_{1j}\sim N(0,\sigma_j^2), j=1,2,\ldots,n$; $\Delta_{2j}\sim N(1,\sigma_j^2), j=1,2,\ldots,n$; $N(u,\sigma^2)$ is the Gaussian distribution, u is the mean and σ^2 is the variance.

2.5. Genetic algorithm

The genetic algorithm is as follows:

Step 0: Calculate the feasible basis of follower dual problem $\boldsymbol{B}_1, \ldots, \boldsymbol{B}_k$;
Step 1: Generate p_s points $x_i, i=1,2,\ldots,p_s$ in boundary constraint set X uniformly. Substitute each x_i into the explicit expression in the follower level. To get $y(x_i)$, let all points x_i be the population pop(0) whose size is p_s. Calculate the fitness of each individual and sort the fitness in ascending order. For conscience, denote pop$(0)=\{x_1,x_2,\ldots,x_{p_s}\}$, let $k=0$;
Step 2: p_c is the crossover probability. Get crossover individual x from pop(k), $\bar{x}$ is the crossover offspring of; $O1$ is set of all crossover offspring;
Step 3: For each individual x in pop(k), mutate with probability p_m, get mutation offspring $\hat{x}$. $O2$ is set of mutation offspring;
Step 4: Calculate fitness of all offspring individual, get $N_1(N_1<p_s)$ best individuals from pop$(k)\cup O1\cup O2$. Get the next generation population pop$(k+1)$ from other p_s-N_1 individuals;
Step 5: If match terminating condition, algorithm end, else let $k=k+1$, go to Step 2.

2.6. Convergence analysis

Definition 1. $\{\xi_i\}$ is a vector sequence of probability space. If ξ exists, prob$\left(\lim\limits_{m\to\infty}\xi_m=\xi\right)=1$, or $\forall\,\varepsilon>0$, prob$\left(\cap_{m=1}^{\infty}\bigcup\limits_{k\ge m}\{\|\xi_k-\xi\|\ge\varepsilon\}\right)=0$. $\{\xi_i\}$ converge to ξ on probability 1.