AN INTEGRATED MODEL OF PARALLEL PROCESSING AND PSO ALGORITHM FOR SOLVING OPTIMUM HIGHWAY ALIGNMENT PROBLEM

Seyed Farzan Kazemi and Yousef Shafahi
Department of Civil Engineering
Sharif University of Technology
Azadi St., Tehran 11365-9313, Iran
Email: shafahi@sharif.edu

KEYWORDS
Optimum Highway Alignment, Parallel Processing, Synchronous and Asynchronous Parallelization, Particle Swarm Algorithm, Master-Slave Architecture.

ABSTRACT

Optimum highway alignment is among the most substantial, but large and complicated topics in transportation area. Infinite number of feasible solutions, numerous local optima and the constrained feature of the problem, associated with complex and mainly non-linear constraints, has put an extra effort into the problem solving process. This paper focuses on solving highway alignment optimization problem using an integrated model of parallel processing and particle swarm optimization algorithm. To achieve this goal, algorithm parallelization is done in synchronous and asynchronous manner. For assessing parallel performance, corresponding indexes are evaluated. SRTM3 databank is used for solving real-world problems. The penalty function approach is employed for dealing with constraints. The successful application of the model is investigated on two real-world route location problems.

INTRODUCTION

Optimum highway alignment was a cumbersome effort during years. In this respect, many researches have been conducted on automatic route locating and design of highways. In the literature, a considerable number of studies have focused on automatic highway alignment design and many people were active in this area.

Designing optimum highway alignment is the most important part of a highway project, since little changes in alignment design leads to considerable changes in the overall cost of construction and even the maintenance of the project. In highway design, engineers nominate some alternatives as initial routes and try to improve these ones by trial and error, based on design constraints and especial project conditions. Finally, an alternative is picked out and finalized based on engineering justice and economic assessment. Noting that the quality of the selected route is highly dependent on the designer's experience, this method cannot be considered as an exact one. Also, if initial routes were not selected on a right basis, high expenditures will be forced to the project, which cannot be compensated even by using best methods for designing project line.

Choosing the final scheme in a highway project is based mainly on economic considerations. In the problem formulation, all sensitive costs should be taken into account. Cost formulation is a tough process which cannot be done in an exact way. For choosing the final alignment, economical comparison between alternatives may be required. The design is usually done separately for horizontal and vertical alignments. The objective of horizontal alignment design is to find the best route, from the economical point of view, to minimize pavement and user costs. In this respect, curvature and inclination of the route is restricted based on the highway type. In vertical alignment, fitting the project line on the natural ground level is investigated. Earthwork is the most costly component in vertical alignment optimization. There have been lots of studies regarding a model which minimizes cut and fill costs. In sophisticated models, other components of cost, such as pavement cost, right of way cost and user cost are also considered.

Not so much far away, choosing optimum highway alignment was done manually, mainly because of computation constraints and lack of mathematical models. By evolving mathematical models and development of high performance computers, highway engineers attempted to improve the design process by employing exact and practical methods. However, current highway alignment design methods are still manual to a large extent. As a result, finding optimum

Proceedings 27th European Conference on Modelling and Simulation ©ECMS Webjørn Rekdalsbakken, Robin T. Bye, Houxiang Zhang (Editors)
ISBN: 978-0-9564944-6-7 / ISBN: 978-0-9564944-7-4 (CD)

highway alignment using automatic computational methods would have satisfying benefits.

It should be noted that optimum highway alignment problem is a multi-modal problem with continuous search space, which may contain local optimums. Since the problem is typically of a large size, meta-heuristic methods are the only available approach, in order to solve it. To do so, choosing sufficient initial answers is required to reach the global or near-global optimum. Moreover, the computer program compiled on the basis of these algorithms must run adequately to gain satisfying levels of reliability. This process is accompanied by so much difficulty and is time-consuming due to the huge amount of computations and the complexity of the problem, resulting use of parallel processing for decreasing the problem solving time. In solving highway alignment problem, parallel processing may help to decrease the problem solving time and make it feasible.

In this study, the particle swarm method is employed to solve optimum highway alignment problem. Based on the large size of the problem and constraints in the geometric design, which shifts the problem into a constrained one, the idea of using parallel processing is utilized to increase the speed of problem solving and decrease the runtime.

A highway design problem is choosing the economical route between two cities (or two points on the map) which connects them to each other. Even in designing a network, the problem is mitigated into finding the optimum alignment connecting each two successive nodes. In general, the problem of designing a transportation network with more than one route can be converted to decompose the whole network into couple pairs and find the optimum route between each of the two.

LITERATURE REVIEW

There have been lots of studies regarding optimum vertical and horizontal alignments. However, the studies focused more on vertical alignment rather than horizontal alignment due to less cost components in vertical alignment. Furthermore, the problem of recursive routes is not considered for vertical alignment.

One of the first researchers who paid attention to optimizing vertical alignment was Easa (Easa 1988). He proposed a model in which project line should make a balance between cut and fill sections. Easa considered longitudinal gradients and vertical curvature as constraints. In next years, Moreb (Moreb 1996) followed Easa's research by reducing the dimensions of the problem. However, the main difference between the two was that Moreb's model guarantees finding global optimum and is more efficient in terms of computational effort.

Horizontal optimization models were developed gently, due to their complex estimation of cost components. Turners and Miles (Turners and Miles 1971) were among the first ones who proposed a horizontal optimization model based on shortest route, using plaid network. In fact, the optimization problem was converted into the network design problem.

Unlike plenty models which optimize horizontal or vertical alignment separately, development of models optimizing both horizontal and vertical alignments simultaneously has been too slow. It may be because of the complication of dealing with three-dimensional space. However, Chaw et al. (Chew et al. 1988) were among the first researchers who figured out the issue. They employed numerical search in their three-dimensional optimization model. Furthermore, Jong and Schonfeld (Jong and Schonfeld 2003) presented an evolutionary model for highway alignment optimization, in which cost components and design constraints are included comprehensively. Their model is capable of dealing with complex and non-derivative objective functions. Additionally, parabolic curves and simple circular curves are designed for vertical and horizontal alignments, respectively.

As increasing development of evolutionary methods, Shafahi and Bagherian (Shafahi and Bagherian 2012) used a PSO algorithm based model for optimizing both vertical and horizontal alignments, simultaneously. A penalty function approach was employed while facing geometric design constraints.

Searching the literature about the utilization of parallel processing in solving optimum highway alignment problem yielded no results. However, this approach is applied widely in other branches of science like Aerospace, Biomechanics and Computer Science.

PARALLEL PROCESSING

Parallel processing is a method of increasing performance through reducing runtime. This procedure is done by distributing subtasks between processors. In general, the goal of using such an approach might be a reduction in the runtime, achieving more accurate responses and economy by simulating intricate systems where simulation is not feasible without parallel processing (Rauber and Runger 2010).

In this study, *Speed-up factor* and *parallel efficiency* are the two parameters which we will use to assess parallelization performance.

PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) algorithm was first developed by Kennedy and Eberhart (Kennedy and Eberhart 1995). This algorithm is inspired by social behavior of bird flocks and schools of fishes and is based on an iterative process. Each particle is a feasible solution of the problem in the search space and each iteration leads to an update in the position of particles, which is done by the following formula:

$$\boldsymbol{x}_i(t+1) = \boldsymbol{x}_i(t) + \boldsymbol{v}_i(t+1) \tag{1}$$

Where:

$\boldsymbol{x}_i(t+1)$ =Position vector for the *i*-th particle at iteration $t+1$;

$\boldsymbol{x}_i(t)$ = Position vector for the *i*-th particle at iteration t;

$\boldsymbol{v}_i(t+1)$ = Velocity vector for the *i*-th particle at iteration t+1

A modified version of this algorithm in which $\boldsymbol{v}_i(t+1)$ is obtained using an inertia factor is proposed by Eberhart and Shi (Eberhart and Shi 1998). In this version, $\boldsymbol{v}_i(t+1)$ is calculated using the following formula:

$$v_{ij}(t+1) = \omega . v_{ij}(t) + c_1 r_1 (p_{ij}(t) - x_{ij}(t)) + c_2 r_2 (p_{gj}(t) - x_{ij}(t)) \tag{2}$$

Where:

$v_{ij}(t+1)$ = The *j*-th dimension of the *i*-th particle's velocity vector at iteration t+1;

ω=Inertia;

$v_{ij}(t)$ = The *j*-th dimension of the *i*-th particle's velocity vector at iteration t;

c_1= Personal cognitive factor;

c_2= Social cognitive factor;

r_1, r_2 = Random numbers lie in the interval [0, 1];

$p_{ij}(t)$ = The *j*-th dimension of the *i*-th particle's best fitness so far at iteration t (Known as *Particle Best*);

$p_{gj}(t)$= The *j*-th dimension of the swarm's best fitness so far at iteration t (Known as *Global Best*)

In this study, inertia (ω) is changing linearly from 0.6 to 0.1 during iterations. Furthermore, personal and social cognitive factors are assumed to be 2.

It worth mentioning that $\boldsymbol{p}_i\ (t)$ and $\boldsymbol{p}_g\ (t)$ are updated at the end of iterations.

Algorithm Parallelization

In every iteration of PSO algorithm, each particle moves to its new position, updates its velocity, updates its own new fitness, updates its personal best and finally, the global best of the swarm is updated. All of these tasks, except the latter one, could be done independently for each particle. This is the main idea of parallelizing the PSO algorithm. By assigning particles to processors, all the tasks, other than updating global best, is done independently on each processor. Only evaluating the global best of the swarm should be done using the output of all processors and cannot be done independently for each single one. In other words, $\boldsymbol{p}_{gj}(\boldsymbol{t})$ in (2) is calculated based on information derived from output of all processors which is determined in the previous iteration. Parallelization in this manner is called *Synchronous Parallelization*.

A practical approach for speeding up parallelization is replacing $\boldsymbol{p}_g(\boldsymbol{t})$ in velocity update equation with $\boldsymbol{p}_g$, which refers to the best particle found in the swarm *so far*. Parallelization in this manner is called *Asynchronous Parallelization*. In the meantime, each particle attaining a better fitness would update the global best and synchronizing operation which was done at the end of each iteration in synchronous parallelization, is neglected. Despite this little modification, it is likely for both parallelization types to lead to almost same results; although calculation details would be different between the two.

Processors' Architecture

For implementing parallel algorithms, a master-slave architecture based on Message Passing Interface (MPI) is employed. In the defined architecture, the master processor doesn't have any specific computational role and only makes communications between the slave ones. For making use of this architecture, after generating initial particles and assigning them to the processors, fitness is evaluated at the end of any iteration through slave processors and is then transmitted to the master processor. The master processor updates the global best and announces the results to the slave processors.

PROCEDURE OF MODELING THE PROBLEM

For modeling the problem, there are three important issues which should be investigated: generating initial particles (paths), dealing with constraints and evaluating cost components.

Generating Initial Paths

To generate an initial path, a random-based algorithm is developed. The pseudo code for this algorithm is presented in figure 1. The coordinate systems *xoy* and *x'oy'* and are global and local systems, respectively. The results of using this algorithm between cities Tehran and Isfahan (located in Iran) is depicted in figure 2 for the following parameters in the pseudo-code:

$y'_{max} = -100\text{km},\ -50\text{km},\ +50\text{km},\ +100km;\ n = 300;\ c = 0.05;$

$m_0 = \dfrac{4y'_{max}}{n \times pd}$.

Pseudo-code for generating initial paths

procedure Generating Initial Paths

  for k=1$\rightarrow$ i **do**

    input m_0, x_0, y_0

    $pd =$ **sqrt** $((x-x_0)^2+(y-y_0)^2)$

    $d = \dfrac{2y'_{max}}{pd.\frac{n}{2}(\frac{n}{2}-1)} - \dfrac{2m_0}{(\frac{n}{2}-1)}$

    for j=1$\rightarrow$ n **do**

      $x' = (j \times pd) + (0.5 - rand()) \times pd$

      if $j \le \frac{n}{2}$ **then**

        $m = m + d$

        $y' = y' + m \times pd$

      else

        $m = m - d$

        $y' = y' - m \times pd$

      end if

      $y' = y' + c \times (1 - 2rand()) \times y'$

      $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$

      Read z from database for each x,y.

      Save *(x,y,z)* as an intersection point of the route.

    end for

  end for

end procedure

Figure 1: Pseudo-code for generating initial paths

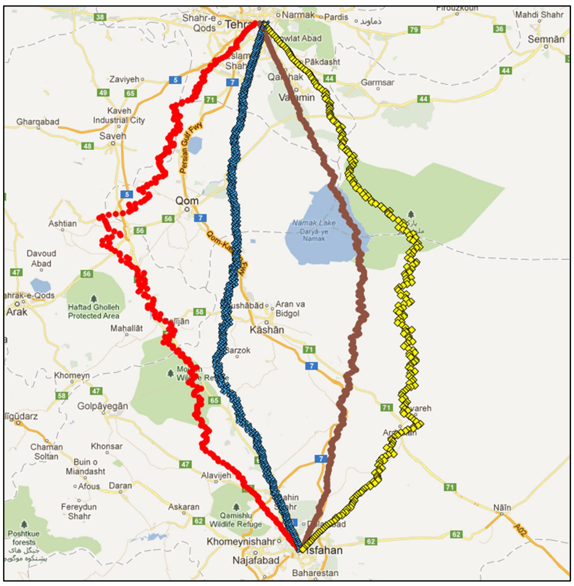

Figure 2: Generating initial paths between cities Tehran and Isfahan (Both cities are located in Iran)

Maximum Curvature of Horizontal Curves

In horizontal alignment, radius of curvature is the most important factor. AASHTO (AASHTO 2004) proposed the minimum permissible radius of horizontal curves as follows:

$$R_{min} = \frac{V^2}{127(0.01e_{\max} + f_{\max})} \tag{3}$$

Where:

$R_{\min}$= Minimum permissible radius, meters;

V= Design speed, km/h;

e_{max} = Maximum allowed superelevation, percent;

$f_{\max}$= Maximum side friction factor

If $L_{i,i+1}$ is the direct distance between intersection points i and i+1 , and Δ_i and Δ_{i+1} are the intersection angles at these points, respectively, one would calculate the minimum radius of curvature between intersection points which prevents the discontinuity of the path, using the following formula:

$$R_{i,i+1} = \frac{L_{i,i+1}}{\tan\frac{\Delta_i}{2} + \tan\frac{\Delta_{i+1}}{2}} \quad \forall i = 1,2,..,\#PI-1 \tag{4}$$

In the latter formula, *#PI* is the number of intersection points. Now we would be able to define a function for calculating penalty function between intersection points i and i+1, which is as follows:

$$C_{H_{i,i+1}} = \begin{cases} a.(\dfrac{1}{\frac{1}{R_{i,i+1}} - \frac{1}{R_{min}}}) & R_{i,i+1} \le R_{min} \\ 0 & R_{i,i+1} \ge R_{min} \end{cases} \quad \forall i = 1,2,..,\#PI-1 \tag{5}$$

In this function, a is a constant which should be tuned based on the nature of the problem in question. The total penalty for route P regarding this constraint can be achieved using the following summation:

$$C_H(P) = \sum_{j=1}^{\#PI-1} C_{H_{j,j+1}} \tag{6}$$

Maximum Longitudinal Grade

The longitudinal grade is penalized, as well as horizontal curvature, if it is greater than maximum allowed grade, which is $g_{\max}$. Considering $g_{i,i+1}$ to be the grade between successive intersection points i and i+1, the penalty function proposed by the authors for exceeding maximum allowed gradient is as follows:

$$C_{G_{i,i+1}} = \begin{cases} b + c.\left(\left|g_{i,i+1}\right| - g_{max}\right) & \left|g_{i,i+1}\right| \ge g_{max} \\ 0 & \left|g_{i,i+1}\right| \le g_{max} \end{cases} \quad \forall i = 0,1,2,..,\#PI \tag{7}$$

In this function, b and c are constants and should be tuned based on the problem type. The penalty cost caused by exceeding maximum grade constraint for path

P would be equal to the following summation:

$$C_G(P) = \sum_{j=0}^{\#PI} C_{G_{j,j+1}} \tag{8}$$

Minimum Length of Vertical Curves

Minimum length of vertical curves is imposed mainly because of satisfying sight distance criteria and is calculated using the following formula, which is proposed by AASHTO (AASHTO 2004):

$$L_{\min} = K.A \tag{9}$$

Where:

$L_{\min}$ = minimum length for vertical curve, meters;
A = Algebraic difference in grades, percent;
K = Rate of vertical curvature, meters

Assuming parabolic symmetric vertical curves for intersection points of the vertical alignment of a highway, the minimum length between intersection points i and i+1 for preventing discontinuity of path is:

$$L_{min_{\,i,i+1}} = \frac{1}{2}(K_i A_i + K_{i+1} A_{i+1}) \tag{10}$$

Again, d is a constant which should be determined empirically. The corresponding penalty function for minimum length of vertical curves is proposed as follows:

$$C_{V_{i,i+1}} = \begin{cases} d.(L_{min_{\,i,i+1}} - L_{i,i+1}) & L_{min_{\,i,i+1}} \geq L_{i,i+1} \\ 0 & L_{min_{\,i,i+1}} \leq L_{i,i+1} \end{cases} \tag{11}$$

So, the total penalty function for path P regarding its minimum length of vertical curves may be calculated through the following summation:

$$C_V(P) = \sum_{j=1}^{\#PI-1} C_{V_{j,j+1}} \tag{12}$$

Cost Evaluation

Cost Components are mainly divided into two major types: *construction costs* and *earthwork costs*. In this study, unit costs are derived from the Iranian Price List for Roads and Runways. The intended construction prices comprise paving costs, base and sub-base courses cost, guardrail costs, curbing costs, lining costs and earthwork costs. Earthworks, including both cut and fill, are calculated using one point cross section and the elevation of the ground points is derived from SRTM3 database, which is partially downloaded into MySQL database.

The Procedure of Modeling the Problem

According to presented descriptions in previous sections, the model for solving the problem is demonstrated in figure 3.

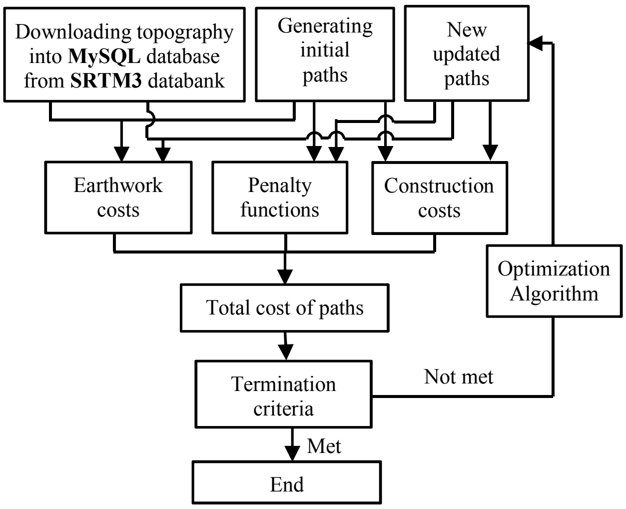

Figure 3: The procedure of modeling the problem

SOLVING REAL WORLD INSTANCES

Two real word route location problem examples are presented and solved based on the proposed model.

First Example: Mashhad-Neyshabur

In the first instance, the optimum highway alignment problem is investigated between the cities Mashhad and Neyshabur, which both are located in Khorasan-e-Razavi province in Iran. Binalood Mountains are located between the two cities and restrict the direct connection of the cities, as can be seen in figure 4. Neyshabur is considered as the origin of the route with coordinates $(36°\ 12'\ 48''\ \text{N}, 58°\ 47'\ 45''\ \text{E})$ and Mashhad is considered as destination which is located at $(36°\ 18'\ 0''\ \text{N}, 59°\ 36'\ 0''\text{E})$. The direct distance between the two cities is 72990 kilometers. The earth range spotted as the search space is a rectangular area 110 kilometers in length and 90 kilometers in width. The parallel processing network used is a network with 4 nodes, each node equipped with two quad-core processors. The program is compiled in Java (version 7) and the operation system used was Linux. The program's constraints are as shown in table 1. The final horizontal intersection points are shown in figure 4. Corresponding parallel performance indexes are presented in table 2. This example is parallelized only in synchronous manner. The stopping criterion was considered 100 iterations.

Second Example: Khoramshahr-Abadan

Finding the optimum highway alignment between the cities Khoramshahr and Abadan was selected as the second example. Khoramshahr, as the origin of the path, is located at $(36°\ 26'\ 22''\text{N}, 48°\ 10'\ 0''\text{E})$ and

Table 1: Design parameters for the first example

	Parameter	Value
Design Parameters	Design speed	110 km/h
	Width of the road	14.6 m
	Slope of cut and fill sections	45°
	Number of intersection points	80
Constraint Parameters	Minimum radius of horizontal curves	450 m
	Maximum allowed longitudinal grade	3%
	Minimum length of crest vertical curves	74A
	Minimum length of sag vertical curves	55A

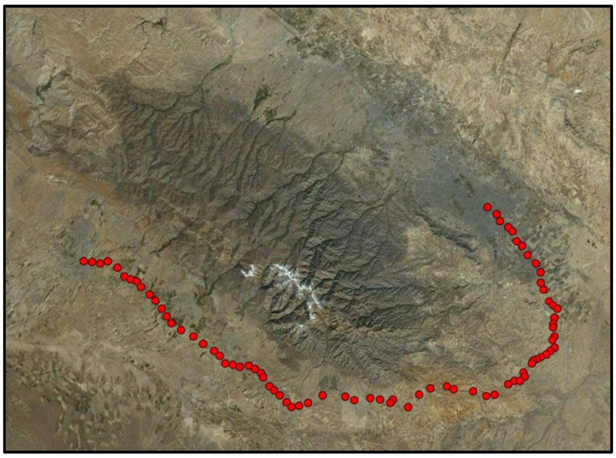

Figure 4: Horizontal alignment of the optimum path for the first example

Table 2: Parallelization indexes for the first example

Number of processors	Runtime (minutes)	Speedup	Parallel efficiency
1	1536	1.00	100.0%
4	403	3.82	95.4%
8	215	7.13	89.1%
16	126	12.19	76.2%
32	84	18.29	57.1%

Abadan, as destination, is located at $(30^\circ\ 21'\ 21''N, 48^\circ\ 18'\ 15''E)$. The reason for choosing theses cities is that these cities lie in a flat region. This is unlike the first example, where the topography was, to a large extent, mountainous. The rectangular search space for this problem is 20 kilometers in length and 15 kilometers in width, as shown in figure 5. Design parameters are similar to the first example, except road width and number of intersection points, which are 7.3 meters and 30, respectively.

This example uses both synchronous and asynchronous parallelization. The synchronous parallelization was accomplished for 100 iterations. The optimum alignment found is very similar to the path

Figure 5: Search space for the second example

which directly connects the origin and destination points due to the plain topography. The result of synchronous parallelization for the vertical alignment is demonstrated in figure 6. Also, parallelization factors are illustrated in table 3.

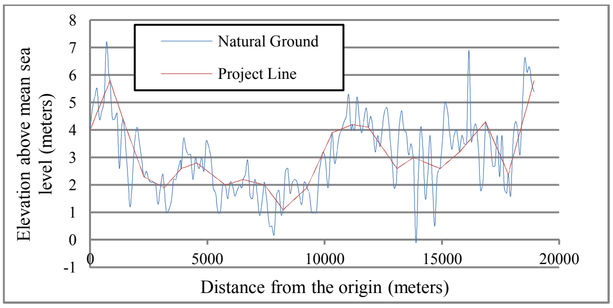

Figure 6: Vertical alignment for the second example (Synchronous parallelization)

Table 3: Synchronous parallelization indexes for the second example

Number of processors	Runtime (seconds)	Speedup	Parallel efficiency
1	5118	1.00	100.0%
4	1344	3.80	95.2%
8	727	7.03	88.0%
16	426	12.01	75.1%
32	291	17.59	55.0%

In asynchronous parallelization, as mentioned before, processors are not necessarily in the same iteration at the same time. In this manner, after completion of an iteration in a processor, the processor enters a new iteration without any delay. Reaching 100 iterations for the first processor was considered as the stopping criterion for this example. It is obvious that the processor with least computational operations will finish its assignments earlier than other processors. It is possible for asynchronous parallelization to have less accurate results, as it is proved by results of this example. The results of applying asynchronous parallelization for reaching the optimum vertical alignment is reflected in figure 7. Furthermore, parallel indexes are presented in table 4.

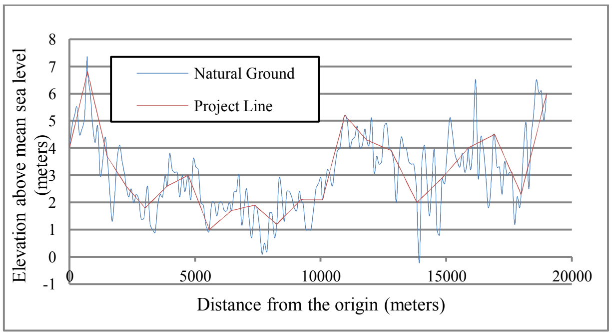

Figure 7: Vertical alignment for the second example (Asynchronous parallelization)

Table 4: Asynchronous parallelization Indexes for the second example

Number of processors	Runtime (seconds)	Speedup	Parallel efficiency
1	3331	1.00	100.0%
4	857	3.82	97.2%
8	416	7.13	95.9%
16	208	12.19	95.1%
32	104	18.29	94.6%

CONCLUSION

In this study, a parallel optimization model was offered to solve optimum highway alignment problem. At the beginning, Particle Swarm Optimization algorithm and two parallel versions of the algorithm were declared. An integrated model of parallel processing and PSO algorithm was then proposed. The application of the offered method on two real world examples was investigated. The first example was a large scale one with mountainous topography. However, the second problem had a flat topography, in order to investigate the efficacy of the model on problems with different types of topography. The second example was solved both synchronously and asynchronously. The results of this example demonstrated that asynchronous parallelization can increase the parallel efficiency considerably, but may have less accurate results. This issue is more likely to occur, especially when the problem in question is large scale. Finally, the results of asynchronous parallelization illustrated that load balancing between processors in solving the problem is of great importance and it is likely to reach less accurate results when applying a poor pattern for distributing particles among processors.

REFRENCES

AASHTO. 2001. *A Policy on Geometric Design of Highways and Streets*. American Association of State Highway and Transportation Officials. Washington, D.C.

Chew, E. P.; C. J. Goh; and T. F. Fwa. 1989. "Simultaneous Optimization of Horizontal and Vertical Alignments for Highways." *Transportation Research Part B*, No.23, 315-329.

Easa, S. M. 1988. "Selection of Roadway Grades that Minimize Earthwork Cost Using Linear Programming." *Transportation Research Part A*, No.22, 121-138.

Eberhart R. C. and Y. Shi. 1998. "A Modified Particle Swarm Optimizer", In *Proceedings of the International Conference on Evolutionary Programming* (Anchorage, AK, May4-9), IEEE, Piscataway, N.J., 69-73.

Jong, J. C. and P. Schonfeld. 2003. "An evolutionary Model for Simultaneously Optimizing Three-Dimensional Highway Alignment." Transportation Research Part B, No.37, 107-128.

Kennedy, J. and R. Eberhart. 1995. "Particle Swarm Optimization." In *Proceedings of IEEE International Conference on Neural Networks* (Perth, WA, Nov.27-Dec.1). IEEE, Piscataway, N.J., 1942-1948.

Moreb A. M. 1996. "Linear Programming Model for Finding Optimal Roadway Grades that Minimize Earthwork Cost" *European Journal of Operational Research*, No.93, 148-154.

Rauber, Th.,and G. Runger. 2010. "Parallel Programming for Multicore and Cluster Systems", Springer, Berlin, Germany.

Shafahi Y. and M. Bagherian. 2012. "A customized Particle Swarm Method to Solve Highway Alignment Optimization Problem." *Computer-Aided Civil and Infrastructure Engineering*, No.28, 52-67.

Turner A. K. and R. D. Miles. 1971. "A Computer-Assisted Method of regional Route Location." *Highway Research Record*, No.348, 1-15.

AUTHOR BIOGRAOHIES

SEYED FARZAN KAZEMI was born in Mashhad, Iran in 1989. He received his Master of Science in Road and Highway Engineering in 2012 from Sharif University of Technology under the supervision of Dr. Shafahi. At this time, he is a doctoral student at University of Tehran. His email address is: s.farzan.kazemi@gmail.com.

YOUSEF SHAFAHI received his Ph.D. in Civil Engineering from University of Maryland, USA in 1997. He joined the Department of Civil Engineering in Sharif University of Technology as an Assistant Professor in 1998 and became an Associated Professor in 2009. His main research interest is operation research in Transportation Planning. His email address is Shafahi@sharif.edu and his web-page can be found at http://sharif.edu/~shafahi.