

Bridges Reinforcement Through Conversion of Tied-Arch Using Crow Search Algorithm

Sergio Valdivia[1,2], Broderick Crawford[1], Ricardo Soto[1], José Lemus-Romani[1(✉)], Gino Astorga[2], Sanjay Misra[3], Agustín Salas-Fernández[1], and José-Miguel Rubio[4]

[1] Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile
{broderick.crawford,ricardo.soto}@pucv.cl,
{jose.lemus.r,juan.salas.f}@mail.pucv.cl
[2] Universidad de Valparaíso, Valparaíso, Chile
{sergio.valdivia,gino.astorga}@uv.cl
[3] Covenant University, Ota, Nigeria
sanjay.misra@covenantuniversity.edu.ng
[4] Universidad Tecnológica de Chile INACAP, Santiago, Chile
jrubiol@inacap.cl

Abstract. The bridges reinforcement is an expensive activity. In this line, new cheap ways to fix bridges have been developed. One of them is the bridges reinforcement through conversion of the cable-stayed arch. The main aim is minimizing the distances between the tensions of each lengthwise position between the original bridge and the bridge with the reinforcement by the cable-stayed arch. For that, it is necessary to know the order that the tension must be applied. In this paper, we resolve this problem by using a recent metaheuristic called Crow Search Algorithm which is based on the intelligent behavior of crows, this approach is inspired by the feature of crows to hide their excess food. The obtained results are compared other approximate techniques by using a well-known statistical test. Promising results reveal that this new algorithm is competitive to solve the proposed problem.

Keywords: Crow Search Algorithm · Metaheuristic optimization · Bridges reinforcement

1 Introduction

In this paper, we will talk about bridges that cross rivers, because that kind of bridges has a very specific problem, the water. It is well know that corrosion [4] and erosion [12] are a big trouble. Also scours can collapse them by the hydraulic action.

Build bridges is an expensive activity [15] and can take years depending on the size. Also, in order to extend the useful life of the structure, this one can be

© Springer Nature Switzerland AG 2019
S. Misra et al. (Eds.): ICCSA 2019, LNCS 11623, pp. 525–535, 2019.
https://doi.org/10.1007/978-3-030-24308-1_42

damaged it is not fixed in the right way. A support plan that allows minimizing expenses is crucial.

Nowadays, new affordable ways to fix bridges structures have been developed by engineering. One of them is the bridges reinforcement through conversion of the cable-stayed arch, the main aim is to install an arch over the bridge and applied tension over hangers that support them, then remove the piers.

Our goal is optimizing those solutions for this new technique can be applied. In order to succeed, we have to minimize the distances between the tensions of each lengthwise position. Besides, we indicate the order that the tension must be applied, because the tension of the cables cannot be performed simultaneously, an excessive tearing can make the bridge collapse (Fig. 1).

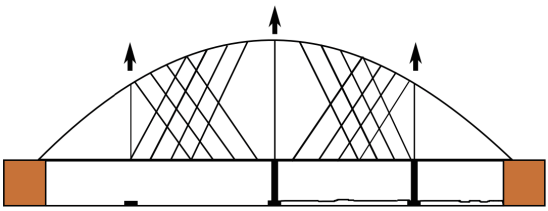


Fig. 1. Bridge reinforcement technique

Finding the best solution for the order of the straps and their magnitude is a complex task that must be supported by some optimization technique. Within the optimization techniques there are complete and approximate ones (Fig. 2) being the metaheuristics an approximate technique that we will use for the solution of the best order and magnitude that we will use.

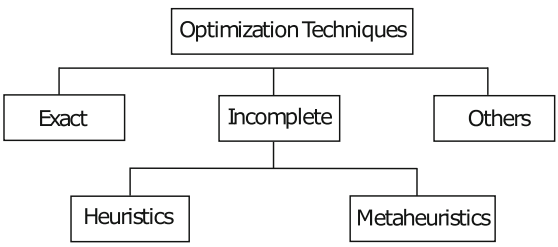


Fig. 2. Optimization techniques

This paper contemplates 5 sections, Sect. 1 is an introduction, Sect. 2 is an explanation of the present problematical, Sect. 3 describes how the new algorithm works, Sect. 4 indicates how we implement the algorithm, also describes the computational results obtained and a statistical test show us comparisons. Finally, Sect. 6 exposes the conclusions of this paper.

2 Problem

Bridges decks are reusable [15]. The concrete of bridges is supported by 3 beams side to side [16]. Beams has finite cuts along lengthwise position that is associated with a tension, we assume work with just 2 beams in order to satisfy the problems because outside beams are similar. Also, the upper and lower tensions of the deck are different, so we must study them because if the maximum or minimum capacity of tension is exceeded, the bridges can collapse. To avoid the collapse is crucial to minimize the distances between the tensions of each lengthwise position [17].

To solve this problem, the metaheuristics will indicate the order and the tensions that must be applied. Subsequently, these solutions are evaluated by the SAP2000 software that makes it possible to design, analyze and build a structure of virtual form and whose communication is realized by means of APIs [1].

2.1 Objective Function

The objective function is defined as the summation of the difference tense of both top and low, for each one of K cuts and each one of the 2 beams, as described by Eq. 1:

$$min \sum_{i=1}^{2} \sum_{k=1}^{k} |\sigma o_{i,k} - \sigma m_{i,k}| \tag{1}$$

Where: $\sigma oi, k$ is the tension of the original bridge in the beam i and in the cut k.
$\sigma m_{i,k}$ is the tension of the bridge modified in the beam i and in the cut k.
This function is evaluated for the low tensions as upper to those of the deck, and the objective is to minimize the differences.

2.2 Constraints

The problem have constraints that must be met to satisfy the objective function

- The hangers cannot be jacking simultaneously.

$$ord_1, ord_2, \ldots, ord_n \in \{1, 2, ..., n\} \tag{2}$$

$$ord_w \neq ord_j \ ; \ \forall w, j \quad con \ w \neq j \qquad w, j \in \{1, 2, \ldots, n\} \tag{3}$$

- The effort of the modified bridge deck should not pass the limits of the Band Admissible Modified (BAM):

$$\sigma m \geqslant \sigma o \tag{4}$$

$$\sigma m \geqslant fct \tag{5}$$

$$\sigma m \leqslant fcmax2 \ (in \ intermediate \ stages) \tag{6}$$

$$\sigma m \leqslant fcmax \ (in \ final \ stages) \tag{7}$$

Where:
σm is the tension (top or bottom) of the modified bridge.
σo is the tension (top or bottom) of the original bridge.
fct is the maximum tension to traction admissible for the concrete.
fcmax is the maximum tension to admissible compression for the concrete.
fcmax2 is the maximum tension to compression for the concrete, extended.
fct, fcmax and fcmax2 are information obtained by SAP2000.

Any tension on the modified bridge deck that is not inside the BAM described from the original model is discarded because can generate damage to the bridge.

3 Algorithm

Engineering and Construction areas are close to optimizations problems, indeed some problems have been solved by heuristics, like Design of reinforced concrete bridge frames [10] or the Infrastructure Asset Management [13]. Additionally, we have antecedents about problems solved by metaheuristics, like Optimal sizing, geometrical and topological design [5] and Discrete optimization of structures [11]. But reinforcement through conversion of cable-stayed arch is almost new in the optimization area. we only have 2 related works the first is proposed by Valenzuela [14] and the second one by Matus [9]. That is the purpose why we are suggesting to test this new algorithm. This type being techniques not only used in problems of this nature [2,8].

3.1 Crow Search Algorithm

Crows are considered like the world's most intelligent animals, some features of this birds are: memorize faces, use tools, communicate in sophisticated ways and hide their excess food in certain positions of the environment and retrieve the stored food when it is needed, Also they have been known to watch other birds, observe where the other birds hide their food, and steal it once the owner leaves.

Assume that at iteration $iter$, crow j wants to visit its hiding place, $m^{j,iter}$. At this iteration, crow i decides to follow crow j to approach to the hiding place of crow j. In this case, two states may happen:

- State 1: Crow j does not know that crow i is following it. As a result, crow i will approach to the hiding place of crow j.
- State 2: Crow j knows that crow i is following it. As a result, in order to protect its hidden place from being stolen, crow j will fool crow i by going to another position of the search space. The new position of crow i is obtained as follows:

$$x^{i,iter+1} = \begin{cases} x^{i,iter} + r_i fl(m^{j,iter} - x^{i,iter}) & r_j \geqslant AP \\ a\ random\ position & \text{otherwise} \end{cases} \tag{8}$$

Where: r_i and r_j are random numbers with uniform distribution between 0 and 1 and fl denotes the flight length of crow i and AP denotes the awareness probability of crow j. Being the movement is applied to each variable.

In Crow Search Algorithm (CSA) [3], intensification and diversification are controlled by the parameter of awareness probability (AP), small values of AP, increases intensification (local search) and large values of AP increases diversification (explore).

Algorithm 1. CSA pseudo code

```
N crows are randomly initialized
Evaluate the position of the crows.
For each crow its memory is initialized.
while iter < iter_max do
    for i = 1 to N do
        A random crow is selected
        for j = 1 to D do
            if r_j ⩾ AP then
                x^{i,iter+1} = x^{i,iter} + r_i fl (m^{j,iter} − x^{i,iter})
            else
                x^{i,iter+1} = a random position of the search space
            end
        end
    end
    Check the feasibility of new positions
    The new position of the crows must be evaluated
    The memory of the crows must be updated
end
```

4 Experimental Results

The first step is take de CSA and make a clear implementation to try to solve this problem, because is very important to prove algorithms in its natural state to verify how they can mutate in time to get better results later.

This problem was proposed by Valenzuela [14] by using Genetic Algorithm (GA) [7] and was compared against Black Hole (BH) [6] by Matus [9]. BH got better results than GA so our experiment will be to compare CSA against BH.

At first we said that the solution vector has the positions of hangers (discrete variables) and the magnitude of tights that must be applied (continues variables), Table 1 show us an example how to represent a solution vector.

Table 1. Example of solution vector

Position 1	Position 2	Position 3	Magnitude 1	Magnitude 2	Magnitude 3
2	1	3	0.78	0.41	0.999

The first hanger that must be tense is position 2 and will have a tension of 41%, then the hanger in position 1 will be tense by 78% and so on, remember that if we have N hangers we will have N magnitudes, for this example we choose a N = 3. As the algorithm explain at the pseudo code the positions of crows that represent a feasible solution is randomly initialize.

4.1 Parameter Settings Used in Experiments

It is necessary to clarify that all instances where executed with the same settings that show us Table 2 for CSA. The values of this configuration were obtained through the parametric scanning technique [8].

Table 2. CSA parameters

Population	Awareness probability	Flight length	Iterations
50	0,4	1e−05	1000

4.2 Computational Results

CSA for Bridges reinforcement was implemented in Python 3.6 and executed in a Personal Computer running Windows 10 on Intel core i7-6700 CPU (3.4 GHz), 16 GB of RAM. The problem has 11 instances, which were executed 15 times.

4.3 Comparison

Now that we have the CSA results we will proceed to compare Fitness against BH proposed by Matus [9].

As we can see at Table 3 the fitness of BHA are compared with CSA in their 15 executions of the instances PV-TCV, HW-TCV, PT-TCV and AB-TCV, at first look we can notice at this moment that BHA is getting better results.

Table 3. Fitness comparation of PV-TCV, HW-TCV, PT-TCV and AB-TC

#	PV-TCV		HW-TC		PT-TCV		AB-TC	
	BH	CSA	BH	CSA	BH	CSA	BH	CSA
1	521950,827	531882,606	518227,336	528989,406	526676,105	530594,126	517068,779	530594,126
2	523427,989	533085,557	518554,722	529211,843	519411,917	531386,203	524023,770	531386,203
3	523381,036	530664,013	521443,876	529494,654	525110,874	530729,813	524821,681	530729,813
4	524362,167	530634,237	518566,592	529751,772	520610,728	530720,987	520848,212	530720,987
5	520927,571	532416,806	516990,684	529558,764	524252,033	530951,184	520622,784	530951,184
6	525788,701	532820,292	522992,527	527809,505	520560,471	531957,290	520152,444	531957,290
7	518788,669	532698,100	522006,562	528984,055	523880,059	532288,375	517564,071	532288,375
8	521114,467	531465,878	519225,675	528889,737	525204,201	530651,000	523623,834	530651,000
9	523983,245	532956,003	520271,778	528398,840	523880,059	531459,352	519373,561	531459,352
10	522809,147	531667,503	523204,076	528122,659	520863,041	533030,705	524246,509	533030,705
11	522648,276	530980,392	520515,557	528762,658	522029,399	530274,053	523785,395	530274,053
12	521351,709	532952,753	519526,112	528202,239	522617,218	531388,957	520203,824	531388,957
13	523941,247	530695,516	517752,373	527747,883	518340,350	530697,510	520872,273	530697,510
14	517407,410	532615,396	521967,482	528308,692	517173,901	530926,098	518973,967	530926,098
15	520202,016	530923,334	521974,260	528870,514	518891,310	531526,666	522447,219	531526,666
Best	517407,410	530664,013	516990,684	527747,883	517173,901	530274,053	517068,779	528454,356

Table 4. Fitness comparation of WR-TCV, VC-TCV, CC-TCV and TC-TCV

#	WR-TCV		VC-TCV		CC-TCV		TC-TCV	
	BH	CSA	BH	CSA	BH	CSA	BH	CSA
1	521210,450	529929,365	517454,836	529169,091	520682,197	527510,958	519824,404	536471,118
2	519448,379	529276,267	516764,747	527600,827	520212,174	527069,873	519571,134	537469,853
3	519880,227	529309,294	526420,203	528470,513	520538,000	526748,922	524103,151	536933,420
4	519204,810	528565,399	523586,259	527963,379	524706,864	527761,352	521301,708	537202,956
5	526646,976	530174,589	517878,923	527091,654	522918,409	528221,432	522386,876	537202,956
6	523896,152	528415,258	521676,411	526987,972	518616,048	527155,806	522834,907	537202,956
7	519162,739	530171,474	515963,267	528123,404	519413,560	528607,198	522191,193	538781,779
8	522611,368	528930,193	524091,123	528572,943	524143,332	527828,843	519926,614	537017,178
9	522236,031	528760,534	520710,801	528714,816	522562,147	527210,139	522591,311	536763,476
10	518685,865	529449,539	520958,228	528606,885	515930,442	528043,516	521634,820	537925,269
11	521471,347	529303,607	522523,255	529405,439	520447,927	527872,354	523052,417	536413,910
12	523417,258	529440,568	521104,325	528034,057	523472,335	527963,324	524988,999	537612,494
13	525384,947	528606,536	521939,460	528436,253	517694,832	526767,144	525268,020	537169,334
14	520876,224	529781,300	521888,839	527493,676	519115,906	527412,494	520641,954	537039,754
15	522051,315	528851,927	520548,389	528610,744	515608,056	529284,467	519598,167	536488,893
Best	518685,865	528415,258	515963,267	526987,972	515608,056	526767,144	519571,134	536413,910

The next step is evaluate instances WR-TCV, VC-TCV, CC-TCV and TC-TCV and again as we see in Table 4 BHA still is getting better results as CSA.

Finally instances RD-AA10, RC-AA10 and CR-AA10 are compared and once again BH got better results as CSA (Table 5).

Table 5. Fitness comparation of RD-AA10, RC-AA10, and CR-AA10

#	RD-AA10		RC-AA10		CR-AA10	
	BH	CSA	BH	CSA	BH	CSA
1	517038,968	523412,188	514019,852	520941,35	517761,806	522137,908
2	519226,401	523972,103	514605,633	522305,13	514470,308	521515,493
3	513572,503	523358,836	517856,008	520951,095	517102,869	521477,768
4	512678,644	522658,448	517790,493	522591,275	517088,897	521148,122
5	511665,241	523982,924	514457,070	520932,178	514502,449	520147,613
6	517200,649	522906,100	514508,111	522357,696	511024,809	520526,858
7	515799,785	523106,237	510072,744	520279,079	516154,932	522624,078
8	516494,402	523035,493	514272,046	521204,525	512943,624	520729,088
9	511107,489	523992,084	515619,911	522056,966	515740,371	520516,490
10	514930,924	521828,964	513847,138	519474,345	512584,970	520488,382
11	509176,298	521545,011	514603,894	519999,895	517126,961	520174,824
12	512632,111	521870,639	513776,522	521028,367	516430,949	521935,361
13	508556,024	522661,916	514564,776	520875,930	514210,370	520867,278
14	520999,995	523603,099	511927,536	522660,726	513743,602	521560,786
15	513408,64	523524,378	513593,241	521046,366	512808,821	520427,006
Best	508556,024	521828,964	510072,744	519474,345	511024,809	520174,824

The Table 6 describe us each instance with the minimum value of fitness of CSA and BH. Also we can see the Relative Change and Difference (RPD) and the difference between the results of the algorithms.

Table 6. Fitness differences and RPD

Instance	Min. CSA	Min. BH	RPD	Difference
PV-TCV	530664,013	517407,410	2,56	13256,603
HW-TCV	527747,883	516990,684	2,08	10757,199
PT-TCV	530274,053	517173,901	2,53	13100,152
AB-TCV	528454,356	517068,779	2,20	11385,577
WR-TCV	528415,258	518685,865	1,88	9729,393
VC-TCV	526987,972	515963,267	2,14	11024,705
CC-TCV	526767,144	515608,056	2,16	11159,088
TC-TCV	536413,910	519571,134	3,24	16842,776
RD-AA10	521828,964	508556,024	2,61	13272,94
RC-AA10	519474,345	510072,744	1,84	9401,601
CR-AA10	520174,824	511024,809	1,79	9150,015

4.4 Instance Distribution

Now that we have all results of both algorithms, we will compare the distribution of the samples of each instance through a violin plot that shows the full distribution of the data Fig. 3.

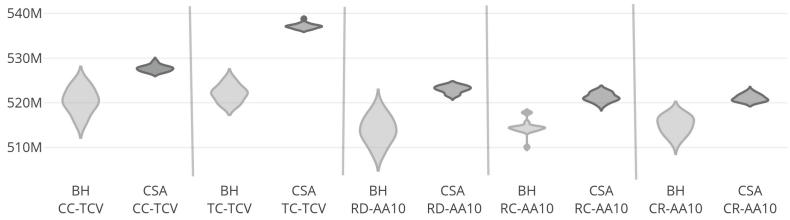


Fig. 3. Instances distribution

After study all the instances we can conclude that all the plots are showing us the same, CSA is caught in a local optimum, do not converge as well as BH.

5 Statistical Tests

It is very important to show how significant difference have BH over CSA in this type of problem, and that is the reason why we perform a contrast statistical test for each instance through the *Kolmogorov-Smirnov-Lilliefors* test to determine the independence of samples and the *Mann-Whitney-Wilcoxon* test to compare the results statistically.

Kolmogorov-Smirnov-Lilliefors test, allow us to analyze the independence of the samples by determining the Z_{min} obtained from the 15 executions of each instance. To determinate independence, we propose the following hypotheses:

- H_0: states that Z_{min} follows a normal distribution.
- H_1: states the opposite.

The test performed has yielded p_value lower than 0.05; therefore, H_0 cannot be assumed. Now that we know that the samples are independent and it cannot be assumed that they follow a normal distribution, it is not feasible to use the central limit theorem. Therefore, for evaluating the heterogeneity of samples we use a non-parametric test call *Mann-Whitney-Wilcoxon* test to compare all the results of each instances we propose the following hypotheses:

- H_0: CSA is better than BH
- H_1: states the opposite.

As we can see at Table 7 BH got better results than CSA in all instances.

Table 7. p-value *Mann-Whitney-Wilcoxon* test

Instance	CSA vs BH	BH vs CSA
PV-TCV	0.999999679	3.21470612e−007
HW-TCV	0.999999679	3.21470612e−007
PT-TCV	0.999999679	3.20839521e−007
AB-TCV	0.999999679	3.21470612e−007
WR-TCV	0.999999679	3.21470612e−007
VC-TCV	0.999999679	3.21470612e−007
CC-TCV	0.999999679	3.21470612e−007
TC-TCV	0.999999681	3.18952541e−007
RD-AA10	0.999999679	3.21470612e−007
RC-AA10	0.999999679	3.21470612e−007
CR-AA10	0.999999679	3.21470612e−007

6 Conclusion

In this paper, bridges reinforcement through conversion of cable-stayed arch by using Crow Search Algorithm, we use a new algorithm to solve the problem of bridges reinforcement, but was very clear after all the tests that CSA is not better than BH solving this problem in its natural state, when we see the graphs of instances distribution we can observe that CSA is trapped in a optimum local. BH always got better results. In order to improve results of CSA it will be necessary to make some modification in the perturbation operator for example or change the way of penalization for the feasible solutions, this options or more will be discuses in a future work, meanwhile Matus et al. [9] still have the best results.

Acknowledgements. Broderick Crawford is supported by Grant CONICYT/ FONDECYT/REGULAR/1171243, Ricardo Soto is supported by Grant CONICYT /FONDECYT /REGULAR/1190129, Gino Astorga is supported by Postgraduate Grant Pontificia Universidad Católica de Valparaíso 2015, José Lemus is Beneficiario Beca Postgrado PUCV 2018. This work was funded by the CONICYT PFCHA/DOCTORADO BECAS NACIONAL/2019 - 21191692.

References

1. Sap2000. http://www.csiespana.com/software/2/sap2000
2. Almonacid, B.: Simulation of a dynamic prey-predator spatial model based on cellular automata using the behavior of the metaheuristic African Buffalo optimization. In: Ferrández Vicente, J.M., Álvarez-Sánchez, J.R., de la Paz López, F., Toledo Moreo, J., Adeli, H. (eds.) IWINAC 2017. LNCS, vol. 10337, pp. 170–180. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59740-9_17
3. Askarzadeh, A.: A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Comput. Struct. **169**, 1–12 (2016)
4. Benin, A.V., Semenov, A.S., Semenov, S.: Fracture analysis of reinforced concrete bridge structures with account of concrete cracking under steel corrosion. Adv. Mater. Res. **831**, 364–369 (2013)
5. Grierson, D.E., Pak, W.H.: Optimal sizing, geometrical and topological design using a genetic algorithm. Struct. Optim. **6**(3), 151–159 (1993)
6. Hatamlou, A.: Black hole: a new heuristic optimization approach for data clustering. Inf. Sci. **222**, 175–184 (2013)
7. Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. MIT Press, Cambridge (1975)
8. Lanza-Gutierrez, J.M., Crawford, B., Soto, R., Berrios, N., Gomez-Pulido, J.A., Paredes, F.: Analyzing the effects of binarization techniques when solving the set covering problem through swarm optimization. Expert Syst. Appl. **70**, 67–82 (2017)
9. Matus, S., Soto, R., Crawford, B.: Optimización del refuerzo de puentes mediante arco atirantado con black hole algorithm. Master's thesis, Escuela de Ingeniería Informática, Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile (2018)
10. Perea, C., Alcala, J., Yepes, V., Gonzalez-Vidosa, F., Hospitaler, A.: Design of reinforced concrete bridge frames by heuristic optimization. Adv. Eng. Softw. **39**(8), 676–688 (2008)
11. Rajeev, S., Krishnamoorthy, C.S.: Discrete optimization of structures using genetic algorithms. J. Struct. Eng. **118**(5), 1233–1250 (1992)
12. Swann, C., Mullen, C.: Predicting erosion impact on highway and railway bridge substructures. Technical report, Mississippi Mineral Resources Institute University of Mississippi and Department of Civil Engineering University of Mississippi, 111 Brevard Hall, Mississippi, June 2016
13. Torres-Machí, C., Pellicer, E., Yepes, V., Chamorro, A.: Heuristic optimization model for infrastructure asset management. In: Bielza, C., et al. (eds.) CAEPIA 2013. LNCS (LNAI), vol. 8109, pp. 300–309. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40643-0_31
14. Valenzuela, M.: Refuerzo de puentes de luces medias por conversion en arco atirantado tipo network. Ph.D. thesis, Universitat Politècnica de Catalunya, Barcelona, España, February 2012
15. Valenzuela, M., Casas, J.: Bridge strengthening by conversion to network arch: design criteria and economic validation. Struct. Infrastruct. Eng. **12**(10), 1310–1322 (2015)
16. Valenzuela, M.A., Casas Rius, J.R.: Bridge strengthening by structural change: optimization via genetic algorithm. In: IABSE-IASS 2011 Symposium (2011)
17. Valenzuela, M.A., Casas Rius, J.R., et al.: Structural behavior and design criteria for bridge strengthening by tied arch. comparison with network arch bridges. In: ARCH 2013, Proceedings of the 7th International Conference on Arch Bridges, pp. 829–836 (2013)