

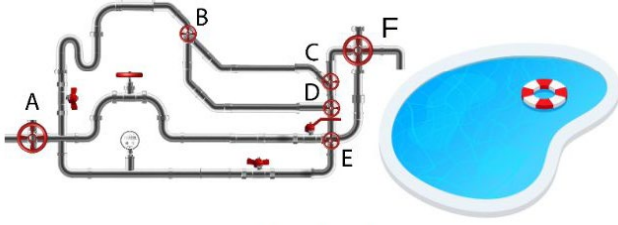
Havuz Problemi

Yasir Erkam ÖZDEMİR

Abstract—Projede, literatürde azami akış (maximum flow) olarak geçen ve düğümler (nodes) arasında akış kapasiteleri belirli bir şekilde (graph) bir başlangıçtan bir hedefe en fazla akışın sağlandığı problemleri çözmemiz beklenmektedir. Devamında ise akışın sistemden geçmemesi için literatürde min-cut olarak geçen yöntemi uygulamamız beklenmektedir.

I. GİRİŞ

Bizden azami akış (maximum flow) olarak bilinen ve düğümler (nodes) arasında akış kapasiteleri belirli bir şekilde (graph) bir başlangıçtan bir hedefe en fazla akışın sağlandığı problemleri çözmemiz beklenmektedir. Sonrasında ise akışın sistemden geçmemesi için literatürde min-cut olarak geçen yöntemi uygulamamız beklenmektedir.



Şekil 1. Örnek bir görsel

Yukarıdaki örnek görselde düğümler muslukları, düğümler arasındaki kenarlar ise musluklar arasındaki boru hatlarının kapasitesini belirtmektedir. Her bir kenar, kenarın izin verebileceği maksimum akış limiti olan ayrı bir kapasiteye sahiptir.

Bizden beklenen A düğümünden F düğüme (daha fazla ya da az miktarda düğüm olabilir) azami miktarda akış sağlayabilmektir.

Ayrıca arayüz üzerinden musluk sayısı, musluklar arasındaki bağlantı bilgisini veren kenarlar ve boru hattının kapasiteleri kullanıcı tarafından alınması gerekmektedir.

Alınan bilgilere göre graf yapısını finamik olarak gösteren bir arayüz hazırlamamız beklenmekte.

Proje çalışırken öncelikle kullanıcı havuzu doldurmaya başlayacak olan başlangıç düğümünü seçecek. Algoritma havuzu maksimum miktarda dolduracak yolu bulacak. Daha sonrasında ise musluktan havuza su akmaması için kesilmesi gereken en az sayıda kenar tespiti yapmamız gerekmektedir. Ayrıca, kesilmesi gereken kenarların hangi noktalar arasında olduğunun yazılması da gerekmektedir.

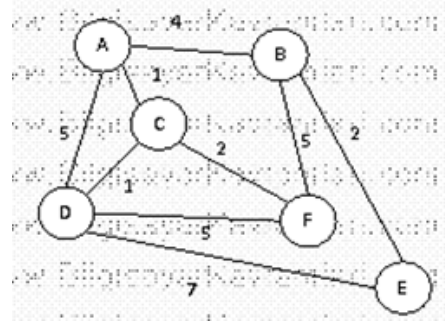
II. YAPILAN ARAŞTIRMALAR

A. Ford Fulkerson Algoritması

Bu algoritmanın amacı, literatürde azami akış (maximum flow) olarak geçen ve düğümler (nodes) arasında

akış kapasiteleri belirli bir başlangıçtan bir hedefe en fazla akışın sağlandığı problemleri çözmektir.

Azami akış (maximum flow) problemini örneğin şehirler arasında bağlı boru hattına veya tedarik zincirine benzetebiliriz. Örneğin aşağıdaki şekli ele alalım:



Şekil 2. Ford fulkerson algoritması örneği

Buradaki düğümler, şehirleri ve düğümler arasındaki kenarlar (edges) ise şehirler arasındaki boru hatlarının kapasitesini belirtsin. Amacımız A düğümünden E düğüme azami miktarda akış sağlayabilmek olsun.

Ford-Fulkerson çözüm için yukarıdaki şekilde öncelikle hedef düğüme giden yolu bulur. Algoritma bu arama işlemi sırasında şayet derin öncelikli arama (depth first search, DFS) kullanıyorsa ford fulkerson olarak isimlendirilir. Şayet aynı algoritma bu arama işlemi sırasında sıfı öncelikli arama (breadth first search, BFS) kullanırsa bu durumda da edmonds karp algoritması olarak isimlendirilir [3].

B. C#

.NET Framework, Microsoft tarafından geliştirilen, açık İnternet protokolleri ve standartları üzerine kurulmuş bir 'uygulama' geliştirme platformu.

C#, yazılım sektörü içerisinde en sık kullanılan iki yazılım dili olan C ve C++ etkileşimi ile türetilmiştir. Ayrıca C#, ortak platformlarda taşınabilir bir (portable language) programlama dili olan Java ile pek çok açıdan benzerlik taşımaktadır. En büyük özelliği ise .Net Framework platformu için hazırlanmış tamamen nesne yönelimli bir yazılım dilidir. Yani nesneler önceden sınıflar halinde yazılıdır. Programcıya sadece o nesneyi sürüklemek ve sonrasında nesneyi amaca uygun çalıştıracak kod satırlarını yazmak kalır.

Microsoft tarafından geliştirilen C#, C++ ve Visual Basic dillerinde yer alan tutarsızlıkları kaldırmak için geliştirilmiş bir dil olmasına rağmen kısa süre içerisinde nesne yönelimli dillerin içinde en gelişmiş programlama dillerinden biri olmayı başarmıştır.

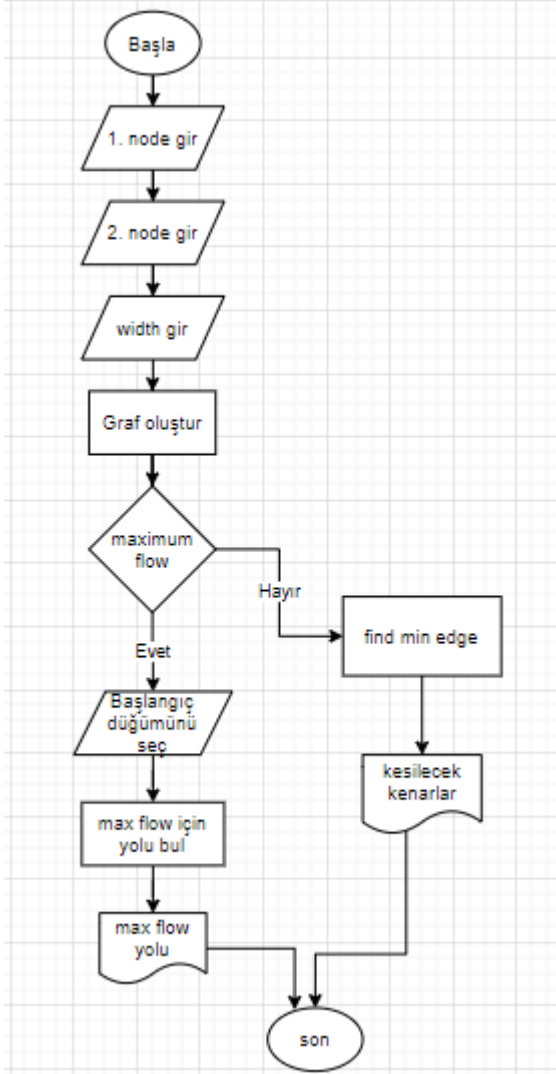
Ayrıca gelişmiş derleyicisi (debugger) ile hata olasılığını ortadan kaldırmaktadır. Yazılan program çalıştırıldıktan sonra derleyici tarafından algılanan Sınıf (Class) ve söz dizimi (syntax) hataları yazılımcıya ayrı bir ekranda ayrıntısı

ile gösterilir ve yazılımcı bu hata penceresinden hataları tespit ederek kolayca düzeltebilir.

Ayrıca pek çok programcı tarafından kullanılan bir dil olmasından ötürü takıldığımız noktalarda uzman programcılardan yardım almak oldukça kolaydır [11].

III. TASARIM

A. Akış Şeması

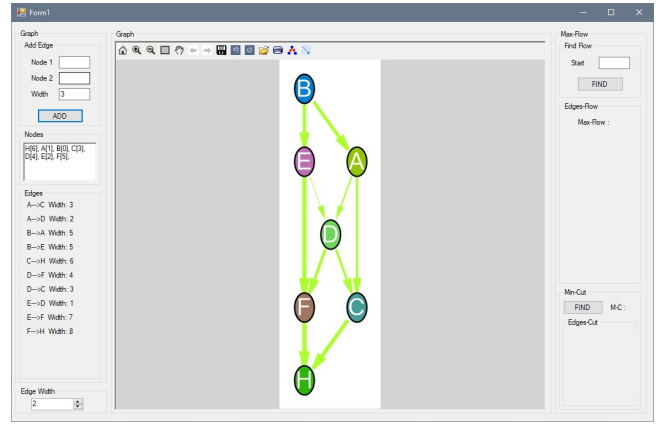


B. Yazılım Mimarisi

Programlama dili olarak C# kullanıldı. Algoritma olarak Ford Fulkerson algoritması kullanıldı.

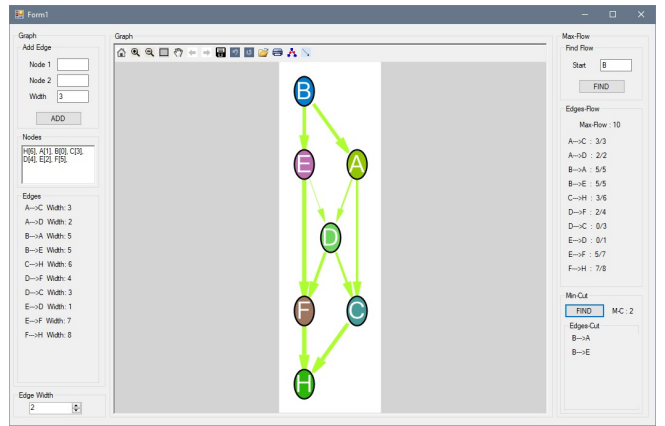
IV. GENEL YAPI

Uygulama açıldıktan sonra ilk olarak kullanıcının musluk sayısını, yani musluk arasında bağlantı bilgisini veren boru hattını (hangi musluklar arasında bağlantı kurulacağını) ve bu boru hattının genişliğini girmesi gerekir.



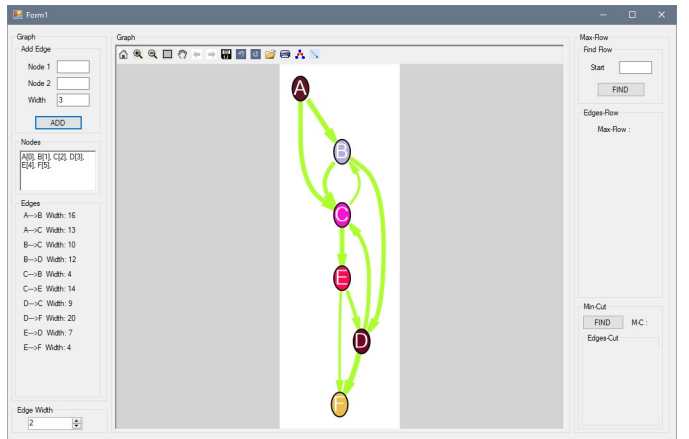
Şekil 3. Uygulama arayüzü

Yukarıdaki şekilde kullanıcının girmesi gereken Node1 Node2 ve width verileri sol üst tarafta gözükmemektedir. Alt kısımda da veriler girildikten sonra ilişki kurulan düğümler arası kenarlar ve bu kenarların genişlikleri gösterilmektedir.

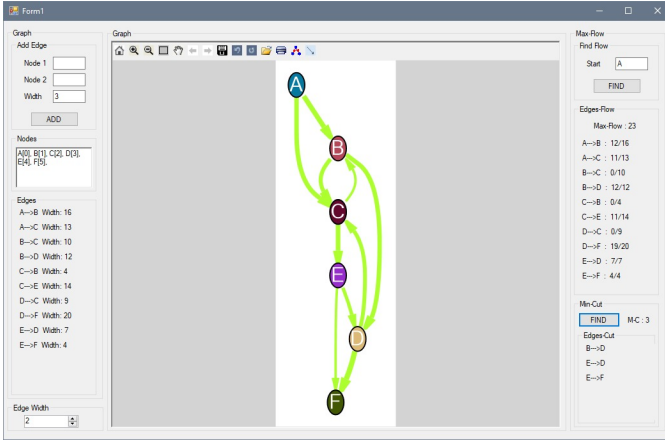


Şekil 4. Max-flow işlemi sonrası ekran görüntüsü

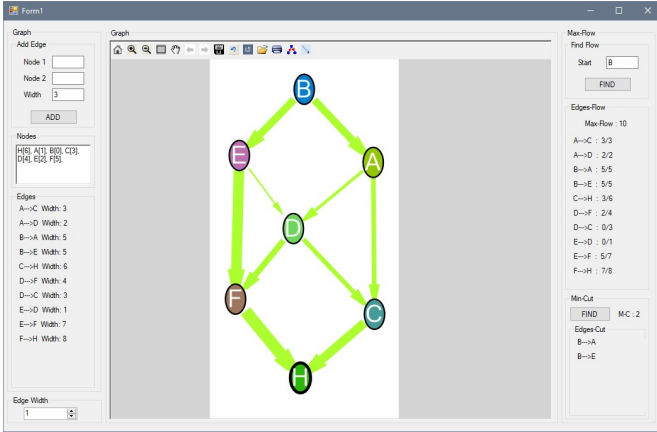
Yukarıdaki resimde de maximum flow işlemi sonrası algoritmanın havuzu maksimum kapasitede dolduracak yolu sağ tarafta gösterilmektedir.



Şekil 5. Uygulama arayüzü



Şekil 6. Max-flow işlemi sonrası ekran görüntüsü



Şekil 7. Uygulama arayüzü

V. REFERANSLAR

- [1] <https://www.microsoft.com/en-us/research/project/microsoft-automatic-graph-layout/?from=http%3A%2F%2Fresearch.microsoft.com%2Fen-us%2Fprojects%2Fmsagl%2F> (Access date: 07.05.2020)
- [2] <http://bilgisayarkavramlari.sadievrenseker.com/2010/05/22/ford-fulkerson-algoritmasi/> (Access date: 22.05.2020)
- [3] https://www.youtube.com/watch?time_continue=1&v=zSeqO7Eno30&feature=emb_title (Access date: 27.05.2020)
- [4] <https://www.geeksforgeeks.org/ford-fulkerson-algorithm-for-maximum-flow-problem/> (Access date: 07.05.2020)
- [5] <https://www.geeksforgeeks.org/minimum-cut-in-a-directed-graph/> (Access date: 07.04.2020)
- [6] <https://github.com/Microsoft/automatic-graph-layout> (Access date: 05.05.2020)
- [7] https://en.0wikipedia.org/wiki/Edmonds%E2%80%93Karp_algorithm (Access date: 08.05.2020)
- [8] <https://gist.github.com/Eyas/7520781#file-edmondskarp-cs-L43> (Access date: 08.05.2020)
- [9] https://en.0wikipedia.org/wiki/Maximum_flow_problem (Access date: 07.05.2020)
- [10] <https://docs.microsoft.com/tr-tr/visualstudio/cross-platform/using-visual-studio-tools-for-unity?view=vs-2019> (Access date: 08.04.2020)
- [11] <https://teknokoliker.com/2011/11/c-nedir-c-temelleri-nelerdir.html> (Access date: 07.05.2020)
- [12]