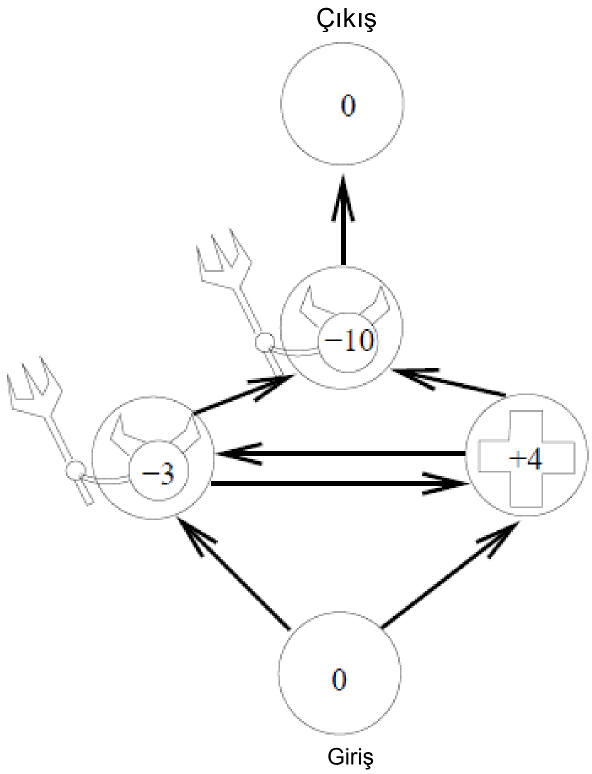

Şekil 3 1-girilebilir labirent örneği.

Profesör Cloud canavarları ve hayat iksirlerini labirente rastgele yerleştirmek için bir program tasarlıyor, ama bazı labirentlerde girişten çıkışa gitmek $L_0 > 0$ olmak zorunda olduğundan mümkün olmayabiliyor. s' den t' ye giden yolda eğer oyuncu sağ kalabiliyorsa bu yol girilebilirdir.
Eğer $L_0=r$ ile başlayan bir labirentte girilebilir bir yol varsa, r-girilebilir labirenti tanımlayın.

Profesöre r"nin en küçük değerini tanımlamak için verimli bir algoritma tasarlamasında yardımcı olun. Öyle ki, bu algoritma, verilen labirentin r-girilebilir olduğunu veya böyle bir r olmadığını belirlesin. (Gidiş yolundan puan almak için, bir labirentin verilen bir r için r-girilebilir olup olmadığı problemini çözün.)

(a) Şekil 3"teki labirent için güvenli bir yol bulun.

(b) Ağırlıkların kenarlarda olduğunu düşünerek soruyu denk bir probleme çevirin, ve bu denkliği kanıtlayın.

Çözüm: Eğer giriş düğümü bir ağırlığa sahipse, 0 giriş ağırlıklı yeni bir giriş yaratır ve bu girişten orijinal girişe bir kenar tanımlarız. Daha sonra, köşelerdeki ağırlıkları kenarlara taşırız. Eğer v düğümü f(v) ağırlığına sahipse, bütün (u,v) Є E"ler için *w(u, v) = f (v)* işlemini yaparız. Denk olan problem kenar ağırlıklı bu grafikte, her alt yolun artı olduğu bir yol bulmadır.

(c) Bu problem için hiç bir hayat puanı arttıran devre olmadığını varsayın. Verilen bir r için, labirentin r-girilebilir olup olmadığını nasıl kontrol edersiniz?

Çözüm: Bunun için Bellman-Ford‟un değiştirilmiş bir sürümünü kullanırız. Verilen bir r için, her bir u düğümü için, oyuncunun u‟ya erişmesi için sahip olacağı en fazla q[u] puanı bulunur. Eğer q[t] artı bir değer ise grafiğimiz r-güvenilirdir.

u∈V olan her köşe için, q[u]‟nun alt sınırı p[u] hesaplanır. Giriş hariç bütün p[u]‟ları başlangıçta -∞‟a atarız. Bellman-Ford algoritmasını koşturarak ve kenarları rahatlatarak p[u] değeri q[u]‟ya yakınsayana kadar artar. Burada, eksi değerli bir düğüme ulaşmanın, bu düğüme ulaşamamaktan iyi olması önemlidir. Yani, eğer p[u] artı oluyorsa onu değiştiririz, aksi takdirde, -∞ olarak saklarız. Rahatlama rutinini aşağıdaki gibi değiştiririz.

```
V-RELAX(u, v)
if (u, v) ∈ E
    then if ((p[v] < p[u] + f[v]) and (p[u] + f[v] > 0))
    then p[v] ← p[u] + f[v]
π[v] ← u
```

Bütün kenarlar V defa rahatlatıldıktan sonra, eğer eksi ağırlıklı hiçbir devre yoksa, bütün p[u]‟lar karşılık gelen q[u]‟ya yakınsar.(En u köşesine erişirken sahip olunan en fazla puanların sayısı). Bu noktada eğer q[t] artı değerli ise, oyuncu çıkışa artı hayat puanları ile ulaşabilir ve grafik r-güvenilir olur.

(d) Şimdi, hayat puanlarını arttıran bir devre olabileceğini varsayın. Verilen bir r için, labirentin r-girilebilir olup olmadığını nasıl kontrol edersiniz?

Çözüm: Eğer p[t] artı değerli değil ise, bütün kenarları bir kere daha rahatlatırız. (Tıpkı Bellman-Ford‟da olduğu gibi). Eğer herhangi bir düğümün p[u]‟su değişirse, r puan ile s‟den ulaşılabilen artı ağırlıklı bir devre bulduk demektir. Yani, oyuncu devrede t‟ye ulaşmak için gerekli olan puana erişebilecek şekilde yeterli defa devrede dolaşır. Eğer artı ağırlıklı bir devre bulamazsak, ve p[t] -∞ ise, grafiğimiz r-güvenilir değildir. Algoritmanın doğruluğu Bellman-Ford‟un doğruluğu ile aynıdır ve koşma süresi O(V E)‟dir.

(e) r-girilebilir bir labirentte, en küçük r değerini nasıl bulabilirsiniz.

Çözüm: Yukarıda verilen alt yöntemi kullanarak en küçük r‟yi buluruz. Önce grafiğin 1-güvenilir olup olmadığını kontrol ederiz. Eğer öyle ise 1‟i cevap olarak döndürürüz. Eğer değilse, 2 ve 4‟ü kontrol ederiz. i‟inci basamakta 2^{i-1}‟i kontrol ederiz. En sonunda 2^{k-1}‟in güvenilir olmadığı ama 2^k‟nın olduğu bir k buluruz. Yani, r‟nin en küçük değeri iki değer arasındadır. Bundan sonra $r=2^{k-1}$ ile $r=2^k$ arasında r‟nin tam değerini bulmak için ikili arama yaparız.

Analiz :

$k = \lfloor \lg r \rfloor$

iken, yapılan işlemlerin sayısı k + O(lgr)=O(lgr)dir. Bellman-Ford‟u O(lgr) sürede koşturmak zorunda olduğumuzdan toplam koşma süresi O(V E lgr)‟dir.